



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**ESTUDIO COMPARATIVO DE PLATAFORMAS PARA ELABORACIÓN DE  
INFORMES EN APLICACIONES WEB. CASO PRÁCTICO: DYNAMIC  
REPORT GENERATOR DE SERIVA S.A.**

**TESIS DE GRADO**

**Previa a la obtención del título de**

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**Presentado por:**

**MERCY PAULINA FALCONÍ CÁRDENAS**

**RIOBAMBA – ECUADOR**

**2009**

## **AGRADECIMIENTO**

A la Ing. Lorena Aguirre y al Ing. Jorge Huilca por su total apertura y recomendaciones para culminar con el presente trabajo de tesis.

A la presidencia y gerencia de Seriva S.A., en las personas de los economistas Ignacio Perez y José Garaicoa, por brindarme todas las facilidades laborales para que pueda desarrollar el trabajo investigativo y el caso práctico en la familia de Seriva.

## **DEDICATORIA**

A mi familia, a mis padres y hermanos, por empujarme a terminar con una etapa de mi vida que había postergado por largo tiempo, por su paciencia y apoyo como coparticipes de este proyecto de tesis.

**NOMBRE**

**FIRMA**

**FECHA**

Dr. Romeo Rodríguez  
DECANO DE LA FACULTAD  
DE INFORMÁTICA Y  
ELECTRÓNICA

\_\_\_\_\_

\_\_\_\_\_

Ing. Iván Menes  
DIRECTOR DE LA ESCUELA DE  
INGENIERÍA EN SISTEMAS

---

---

Ing. Lorena Aguirre  
DIRECTORA DE TESIS

---

---

Ing. Jorge Huilca  
MIEMBRO DEL TRIBUNAL

---

---

Lic. Carlos Rodríguez  
DIRECTOR CENTRO DE  
DOCUMENTACIÓN

---

---

NOTA DE LA TESIS

“Yo, Mercy Paulina Falconí Cárdenas, soy la responsable de las ideas, doctrinas y resultados expuestos en esta tesis, y el patrimonio intelectual de la misma pertenecen a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”

---

Mercy Paulina Falconí Cárdenas



## ÍNDICE DE ABREVIATURAS

<b>Ad hoc.</b>	Locución latina que significa literalmente «para esto»
<b>ADO.</b>	ActiveX Data Objects
<b>ALM.</b>	Application Lifecycle Management
<b>APIs.</b>	Application Programming Interfaces
<b>BA.</b>	Business Administrators
<b>BI.</b>	Business Intelligence
<b>BIRT.</b>	Eclipse Business Intelligence Reporting and Tools
<b>CLS.</b>	Common Language Specification
<b>CRM.</b>	Customer Relationship Management
<b>DAO.</b>	Data Access Object
<b>DDL.</b>	Data Definition Language
<b>DOM.</b>	Document Object Model
<b>DRG.</b>	Dynamic Report Generator
<b>DTD.</b>	Document Type Definition
<b>EDM.</b>	Entity Data Model
<b>EJB.</b>	Enterprise Java Bean
<b>ETW.</b>	Event Tracing for Windows
<b>FAST.</b>	Facilitated Application Specification Techniques
<b>GUI.</b>	Graphic User Interfaz
<b>HDML.</b>	Handheld Device Markup Language
<b>HOLAP.</b>	Hybrid OLAP
<b>HSM.</b>	Hardware Security Module
<b>HTL.</b>	High Technology in Learning
<b>I&amp;KM.</b>	Information and Knowledge Management
<b>IIS.</b>	Internet Information Services
<b>ISV.</b>	Independent Software Vendors
<b>JSF.</b>	Java Server Faces
<b>LINQ.</b>	Language Integrated Query
<b>MOLAP.</b>	Mobile OLAP
<b>NTLM.</b>	NT LAN Manager
<b>ODA.</b>	Open Data Access
<b>OGC.</b>	Open Geospatial Consortium
<b>OLAP.</b>	On-Line Analytical Processing
<b>PDA.</b>	Personal Digital Assistants
<b>PEI.</b>	Plataformas para Elaboración de Informes
<b>RDC.</b>	Report Definition Customization
<b>RDCE.</b>	Report Definition Customization Extension
<b>RDL.</b>	Microsoft Reporting services Definition Language
<b>RFID.</b>	Radio Frequency IDentification
<b>ROLAP.</b>	Relational OLAP
<b>RTF.</b>	Rich Text Format
<b>SOAP.</b>	Simple Object Access Protocol
<b>SSRS.</b>	Sql Server Reporting Services
<b>TDE.</b>	Transparent Data Encryption
<b>UDT.</b>	User Defined Type
<b>W3C.</b>	World Wide Web Consortium
<b>WMI.</b>	Windows Management Instrumentation
<b>WML.</b>	Wireless Markup Language
<b>WYSIWYG.</b>	What You See Is What You Get ("lo que ves es lo que obtienes").
<b>XAML.</b>	eXtensible Application Markup Language



# CAPITULO I:

## ÍNDICE GENERAL

<b>CAPITULO I: MARCO PROPOSITIVO.....</b>	<b>17</b>
1.1. ANTECEDENTES .....	17
1.2. JUSTIFICACION .....	19
1.3. OBJETIVOS .....	21
1.3.1. <i>Objetivo General</i> .....	21
1.3.2. <i>Objetivos Específicos</i> .....	21
1.4. HIPÓTESIS .....	21
<b>CAPITULO II: MARCO TEÓRICO .....</b>	<b>22</b>
2.1. INTRODUCCIÓN .....	22
2.2. PLATAFORMA PARA ELABORACIÓN DE INFORMES .....	23
2.3. PRINCIPALES CARACTERÍSTICAS DE UNA PEI.....	23
2.3.1. <i>Tipos de reportes</i> .....	24
2.3.1.1. <i>Reportes Paramétricos</i> .....	24
2.3.1.2. <i>Reportes Enlazados</i> .....	25
2.3.1.3. <i>Reportes de Instantáneas</i> .....	25
2.3.1.4. <i>Reportes Almacenados en Cache</i> .....	26
2.3.1.5. <i>Reportes Ad Hoc</i> .....	26
2.3.1.6. <i>Reportes Clickthrough</i> .....	26
2.3.1.7. <i>Reportes Drillthrough</i> .....	26
2.3.1.8. <i>Subreportes</i> .....	26
2.3.2. <i>Madurez del Producto</i> .....	27
2.3.3. <i>Diseñador de informes</i> .....	27
2.3.4. <i>Licenciamiento</i> .....	27
2.3.5. <i>Formato del Archivo de Reporte</i> .....	27
2.3.6. <i>Acceso a Datos</i> .....	28
2.3.7. <i>Data Binding</i> .....	28
2.3.8. <i>Características de Diseño de Reportes</i> .....	28
2.3.9. <i>Formatos de Exportación</i> .....	29
2.3.10. <i>Manejo de Parámetros</i> .....	29
2.3.11. <i>Arquitectura</i> .....	29
2.3.12. <i>Costos</i> .....	30
2.3.13. <i>Reportes AdHoc</i> .....	30
2.3.14. <i>Gráficas</i> .....	30
2.3.15. <i>Rendimiento</i> .....	30
2.3.16. <i>Escalabilidad</i> .....	31
2.3.17. <i>Seguridad</i> .....	31
2.4. CAPACIDAD DE DEFINICIÓN Y CREACIÓN DE REPORTES .....	31
2.4.1. <i>Capacidad para publicación de reportes</i> .....	31
2.5. PRINCIPALES PEI DEL MERCADO.....	31
2.5.1. <i>SAP Business Objctcs Crystal Reports 2008</i> .....	34
2.5.2. <i>SAS® Web Report Studio</i> .....	37
2.5.3. <i>SQL Server Reporting Services 2008</i> .....	38
2.5.4. <i>Performance Point Server 2007</i> .....	41
2.5.5. <i>SQL 2008 Report Builder</i> .....	42
2.5.6. <i>Access Reports</i> .....	43
2.5.7. <i>Excel 2007 Connecting to Olap Cubes &amp; Relational Databases</i> .....	44
2.5.8. <i>XSL FO</i> .....	45

2.5.9.	<i>Report Sharp –Shooter</i> .....	47
2.5.10.	<i>BinaryWorX BXQuery – ActiveX</i> .....	48
2.5.11.	<i>SBS Development</i> .....	49
2.5.12.	<i>TrendAnalyzer</i> .....	51
2.5.13.	<i>Total Access Memo – ActiveX</i> .....	51
2.5.14.	<i>Aspose AdHoc - .Net</i> .....	52
2.5.15.	<i>ComponentOne Studio</i> .....	53
2.5.16.	<i>DXperience</i> .....	54
2.5.17.	<i>Data Dynamics</i> .....	55
2.5.18.	<i>Clover</i> .....	59
2.5.19.	<i>Enterprise Reporting Applications de Actuate</i> .....	60
2.5.20.	<i>Hyperion System 9 BI+</i> .....	64
2.5.21.	<i>Cognos 8 Business Intelligence, Reporting</i> .....	65
2.5.22.	<i>Oracle reports</i> .....	67
2.5.23.	<i>MicroStrategy Enterprise Reporting</i> .....	69
2.6.	<b>CUADRO DE VALORIZACIÓN DE PEI.</b> .....	71
2.7.	<b>CRYSTAL REPORTS</b> .....	74
2.7.1.	<i>Arquitectura</i> .....	74
2.7.2.	<i>Interfaces de desarrollo</i> .....	75
2.7.3.	<i>Creación de contenidos y acceso a los datos</i> .....	76
2.7.4.	<i>Informes</i> .....	76
2.7.5.	<i>Análisis</i> .....	77
2.7.6.	<i>Repositorio Crystal</i> .....	77
2.7.7.	<i>Características Principales</i> .....	78
2.7.8.	<i>Servicios sobre la plataforma Crystal Decisions</i> .....	78
2.8.	<b>MICROSOFT REPORTING SERVICES</b> .....	78
2.8.1.	<i>Arquitectura</i> .....	79
2.8.2.	<i>Servicios e Infraestructura</i> .....	81
2.8.3.	<i>Autenticación y procesamiento de memoria para procesamiento en Background</i> .....	83
2.8.4.	<i>Reporting Services WMI Provider</i> .....	83
2.8.5.	<i>Novedades en Programación de Report Server</i> .....	84
2.8.6.	<i>Model Designer en Business Intelligence Development Studio</i> .....	84
2.8.7.	<i>Administración de informes Reporting Services en Modo Sharepoint</i> .....	84
2.8.8.	<i>AdHoc reporting</i> .....	85
2.8.9.	<i>Embedded reporting</i> .....	85
2.8.10.	<i>Web-based reporting for partners/customers</i> .....	85
2.9.	<b>OFFICE SYSTEM</b> .....	87
2.10.	<b>TENDENCIAS PARA LA GENERACIÓN DE INFORMES DINÁMICOS</b> .....	87
2.10.1.	<i>Ad Hoc Querying</i> .....	88
2.11.	<b>ANÁLISIS COMPARATIVO ENTRE SSRS 2008 Y SAP BUSINESS CRYSTAL REPORTS 2008</b> .....	89
2.11.1.	<i>Madurez del Producto</i> .....	89
2.11.2.	<i>Licenciamiento</i> .....	89
2.11.3.	<i>Formato del Archivo de Reporte</i> .....	89
2.11.4.	<i>Acceso a Datos</i> .....	90
2.11.5.	<i>Data Binding</i> .....	90
2.11.6.	<i>Diseñador de reportes</i> .....	91
2.11.7.	<i>Características de Diseño de Reportes</i> .....	91
2.11.8.	<i>Formatos de Exportación</i> .....	93
2.11.9.	<i>Manejo de Parámetros</i> .....	93
2.11.10.	<i>Arquitectura</i> .....	94
2.11.11.	<i>Costos</i> .....	94
2.11.12.	<i>Reportes AdHoc</i> .....	94
2.11.13.	<i>Tablix</i> .....	95

2.11.14.	<i>Gráficas</i> .....	95
2.11.15.	<i>Tipos de Reportes</i> .....	95
2.11.16.	<i>Conclusión</i> .....	95
2.12.	CUADRO COMPARATIVO DE SAP BUSINESS OBJECTS VS SQL SERVER REPORTING SERVICES .....	95
2.13.	LA TECNOLOGÍA .NET .....	111
2.13.1.	<i>ASP.NET 3.5</i> .....	112
2.13.2.	<i>LINQ</i> .....	114
2.14.	SERVICIOS WEB Y WINDOWS COMMUNICATION FOUNDATION.....	115
2.14.1.	<i>Características de WCF</i> .....	116
2.14.2.	<i>Arquitectura Orientada a Servicios</i> .....	117
2.14.3.	<i>Composición básica de una aplicación WCF</i> .....	117
2.15.	SERIALIZACIÓN Y DESERIALIZACIÓN DE OBJETOS .....	118
2.15.1.	<i>Selección del Formato de Serialización</i> .....	119
2.15.2.	<i>Serialización usando BinaryFormatter</i> .....	119
2.16.	MICROSOFT VISUAL STUDIO 2008 .....	119
2.17.	MICROSOFT C# 2008 .....	120
2.17.1.	<i>Características del lenguaje C# 2008</i> .....	121
2.17.2.	<i>Variables locales de tipos implícitos</i> .....	121
2.17.3.	<i>Propiedades Automáticas</i> .....	122
2.17.4.	<i>Métodos de Extensión</i> .....	123
2.17.5.	<i>Métodos Parciales</i> .....	124
2.17.6.	<i>Sintaxis de Inicialización de Objetos</i> .....	125
2.17.7.	<i>Inicialización de Colecciones</i> .....	125
2.17.8.	<i>Tipos Anónimos</i> .....	125
2.18.	LINQ APIs.....	128
2.18.1.	<i>LINQ</i> .....	129
2.18.2.	<i>Funcionamiento de LINQ</i> .....	130
2.18.3.	<i>Modelo Relacional vs Modelo Jerárquico/Gráfico</i> .....	131
2.18.4.	<i>Manejo de XML</i> .....	133
2.18.5.	<i>Integración de lenguajes</i> .....	134
2.18.6.	<i>LINQ to Objects</i> .....	135
2.18.7.	<i>LINQ to ADO.NET</i> .....	135
2.19.	MICROSOFT SQL SERVER 2008.....	136
2.19.1.	<i>Visión de plataforma de datos de Microsoft</i> .....	138
2.20.	PARTNERS DE MIGRACIÓN DE CRYSTAL A REPORTING SERVICES .....	139
<b>CAPITULO III:</b>	<b>IMPLEMENTACIÓN DEL SOFTWARE: DYNAMIC REPORT GENERATOR. ....</b>	<b>144</b>
3.1.	INTRODUCCIÓN .....	144
3.1.1.	<i>Presentación</i> .....	145
3.1.2.	<i>Metodología de Desarrollo</i> .....	146
3.1.3.	<i>Visión</i> .....	146
3.1.4.	<i>Ambientes de Operación</i> .....	147
3.1.5.	<i>Referencias</i> .....	147
3.2.	METAS DE DISEÑO .....	147
3.2.1.	<i>Rendimiento</i> .....	147
3.2.2.	<i>Disponibilidad</i> .....	147
3.2.3.	<i>Fiabilidad</i> .....	148
3.2.4.	<i>Seguridad</i> .....	148
3.2.5.	<i>Interoperabilidad</i> .....	148
3.3.	FASE DE PLANIFICACIÓN .....	148
3.3.1.	<i>Definición del ámbito del software</i> .....	148
3.3.2.	<i>Perspectiva de Dynamic Report Generator</i> .....	149

3.3.3.	<i>Características del Usuario</i>	149
3.3.4.	<i>Perfiles de Usuario y Accesos al Sistema</i>	150
3.3.5.	<i>Especificación de requerimientos</i>	150
3.3.5.1.	<i>Especificación de Funciones del Sistema</i>	150
3.3.5.2.	<i>Especificación de Atributos del Sistema</i>	151
3.3.6.	<i>Estimaciones</i>	152
3.3.6.1.	<i>Métricas</i>	152
3.3.6.2.	<i>Costo total del desarrollo de la solución.</i>	154
3.3.7.	<i>Gestión de Riesgos</i>	154
3.3.7.1.	<i>Identificación y proyección del riesgo</i>	155
3.3.7.2.	<i>Evaluación del Riesgo</i>	156
3.3.7.3.	<i>Gestión y supervisión del riesgo</i>	157
3.4.	<b>FASE DE ANÁLISIS</b>	<b>157</b>
3.4.1.	<i>Definición de Casos de uso</i>	157
3.4.1.1.	<i>Identificación de actores</i>	157
3.4.1.2.	<i>Caso de uso general</i>	158
3.4.1.3.	<i>Casos de uso particulares</i>	159
3.4.2.	<i>Diagramas de Casos de uso</i>	162
3.4.2.1.	<i>Diagrama de caso de uso general – Contexto Funcional</i>	162
3.4.2.2.	<i>Diagramas de caso de uso particulares</i>	163
3.4.3.	<i>Construcción del Modelo Conceptual</i>	166
3.4.3.1.	<i>Diagrama conceptual de base de datos</i>	166
3.4.4.	<i>Construcción del Modelo Lógico</i>	168
3.4.5.	<i>Diagrama de Estados</i>	169
3.4.6.	<i>Diagramas de Secuencia</i>	170
3.5.	<b>FASE DE DISEÑO</b>	<b>171</b>
3.5.1.	<i>Diagrama de Colaboración</i>	171
3.5.2.	<i>Diagrama de Clases del Diseño</i>	172
3.5.3.	<i>Diseño de Base de Datos</i>	173
3.5.4.	<i>Diagrama de Componentes</i>	174
3.5.5.	<i>Diagrama de Despliegue</i>	174
3.6.	<b>FASE DE IMPLEMENTACIÓN</b>	<b>175</b>
3.7.	<b>FASE DE ESTABILIZACIÓN</b>	<b>177</b>
3.7.1.	<i>Prueba Piloto</i>	177
<b>CAPITULO IV: PRESENTACIÓN Y ANÁLISIS DE RESULTADOS</b>		<b>180</b>
4.1.	<b>COMPROBACIÓN DE LA HIPÓTESIS</b>	<b>180</b>
4.1.1.	<i>Variables e indicadores</i>	180
4.1.2.	<i>Operacionalización Metodológica</i>	182
4.1.3.	<i>Coefficiente de Correlación de Pearson</i>	183
<b>CONCLUSIONES</b>		
<b>RECOMENDACIONES</b>		
<b>RESUMEN</b>		
<b>SUMMARY</b>		
<b>GLOSARIO</b>		
<b>ANEXOS</b>		
<b>BIBLIOGRAFÍA</b>		

## ÍNDICE DE TABLAS

Tabla II-1. Parámetros generales para la valoración de una PEI .....	32
Tabla II-2. Valoración General Crystal Reports .....	37
Tabla II-3- Valoración General SAS Web Report Studio .....	38
Tabla II-4. Valoración General SSRS .....	41
Tabla II-5. Valoración General Performance Point Server.....	42
Tabla II-6. Valoración General SQL 2008 Report Builder.....	43
Tabla II-7. Valoración General Access Reports.....	44
Tabla II-8. Valoración General Excel 2007 Connecting to Olap Cubes.....	44
Tabla II-9. Valoración General XSL FO .....	46
Tabla II-10. Valoración General Report Sharp –Shooter .....	48
Tabla II-11. Valoración General BinaryWorX BXQuery .....	49
Tabla II-12. Valoración General <i>XL-DBQuery</i> .....	49
Tabla II-13. Valoración General Automated-SQL-Builder.....	50
Tabla II-14. Valoración General TrendAnalyzer.....	51
Tabla II-15. Valoración General Total Access Memo .....	52
Tabla II-16. Valoración General Aspose AdHoc - .Net .....	53
Tabla II-17. Valoración General ComponentOne Studio .....	54
Tabla II-18. Valoración General DXperience .....	55
Tabla II-19. Valoración General ActiveReports for .Net.....	56
Tabla II-20. Valoración General DynamiCube .....	57
Tabla II-21. Valoración General ContourCube.....	59
Tabla II-22. Valoración General Clover.....	60
Tabla II-23. Valoración General Actuate BIRT.....	62
Tabla II-24. Valoración General Hyperion System 9 BI+ .....	65
Tabla II-25. Valoración General Cognos 8 Business Intelligence.....	67
Tabla II-26. Valoración General Oracle reports .....	68
Tabla II-27. Valoración General MicroStrategy Enterprise Reporting.....	70
Tabla II-28. Cuadro de valorización de herramientas.....	71
Tabla II-29. Cuadro comparativo de características CR vs SSRS .....	97
Tabla II-30. Cuadro comparativo de capacidad de definición y creación de reportes CR vs SSRS.....	109
Tabla III-1. Entregables DRG.....	145
Tabla III-2. Perfiles de Usuario y Accesos al Sistema DRG .....	150
Tabla III-3. Ajuste de valores de complejidad .....	152
Tabla III-4. Valores de ajuste .....	153
Tabla III-5. Computación de métricas de puntos de funciones.....	153
Tabla III-6. CU-03.....	154
Tabla III-7. Tabla de riesgos .....	155
Tabla III-8. Categoría de riesgos.....	156
Tabla III-9. Impacto del riesgo.....	156
Tabla III-10. Identificación actores primarios .....	157
Tabla III-11. Caso de uso general.....	158
Tabla III-12. CU-02.....	159
Tabla III-13. CU-03.....	159
Tabla III-14. CU-04.....	160
Tabla III-15. CU-05.....	161
Tabla IV-1. Variables de la Hipótesis.....	181
Tabla IV-2. Operacionalización Metodológica de las Variables.....	182
Tabla IV-3. Cuadro resumen de comparación CR y SSRS .....	184
Tabla IV-4. Cuadro de valores para el cálculo del coeficiente de Pearson .....	187
Tabla 0-1. Significación del coeficiente de correlación de Pearson .....	195

## ÍNDICE DE FIGURAS

Figura I. Tratamiento de documentos en XSL - FO .....	45
Figura II. Promedio de valoración PEIs .....	72
Figura III. Valoración de las herramientas más robustas para elaboración de informes.....	73
Figura IV. Presencia en el mercado de plataformas de reporting más robustas.....	73
Figura V. Arquitectura Funcional de Crystal Reports 2008 .....	75
Figura VI. Arquitectura Funcional de SSRS.....	81
Figura VII. Arquitectura de Servicios SSRS.....	82
Figura VIII. Interoperabilidad de los servicios web XML.....	115
Figura IX. Relación entre las tres capas de WCF en alto nivel .....	117
Figura X. Visión de plataforma de datos de Microsoft SQL Server .....	138
Figura XI. Diagrama de caso de uso general .....	162
Figura XII. CU-02 Definir variables .....	163
Figura XIII. CU-03 Crear Reportes Ad Hoc.....	164
Figura XIV. CU-04 Definir de mecanismos de distribución.....	165
Figura XV. CU-05 Consultar reportes generados .....	165
Figura XVI. Diagrama conceptual de base de datos .....	166
Figura XVII. Diagrama de Clases.....	168
Figura XVIII. Diagrama de Estados.....	169
Figura XIX. Diagrama de Secuencia.....	170
Figura XX. Diagrama de Colaboración .....	171
Figura XXI. Diagrama de clases del diseño.....	172
Figura XXII. Diseño de la base de datos .....	173
Figura XXIII. Diagrama de componentes.....	174
Figura XXIV. Diagrama de despliegue.....	175
Figura XXV. Distribución de la solución DRG .....	176
Figura XXVI. Elementos de la solución DRG.....	177
Figura XXVII. Base de datos de prueba de estabilización.....	178
Figura XXVIII. Datos generales de una variable.....	178
Figura XXIX. Definición de esquema de acceso para una variable.....	179
Figura XXX. Reporte generado.....	179
Figura XXXI. Sobreposición de cacterísticas de la plataforma para la elaboración de informes .....	186
Figura XXXII. Sobreposición capacidad de definición y creación de reportes .....	186
Figura XXXIII. Gráfica de dispersión de las variables de la hipótesis .....	188
Figura XXXIV. Distribución normal del coeficiente de Pearson.....	189

## INTRODUCCIÓN

El activo más valioso de una entidad de cualquier sector, sea comercial, educativo o gubernamental, es su información. Un sistema informático está constituido principalmente por un repositorio de datos, los cuales se ingresan, modifican, o eliminan con apremiante celeridad. Pero la verdadera valía de los datos de un sistema, no es precisamente la sustitución de papel o archivadores, sino representar un medio para la toma de decisiones, ser la herramienta mediante la cual los usuarios puedan realizar estimaciones, mediciones de tendencias, o proyecciones de su actividad.

Es evidente el protagonismo de los datos, y por ello hoy en día disponemos de una gran variedad de software para administración, minería, presentación, publicación, etc., de la información que guarda una aplicación.

Una importante área del desarrollo de sistemas de cómputo, es la implementación de soluciones de reportes. Los reportes son a menudo la salida más visible de una aplicación de software. Submódulos encargados de la presentación de la información a los usuarios en términos comprensibles previstos de flexibilidad para adaptarse a las necesidades de cada persona y de la usabilidad y eficiencia requeridos para la satisfacción de los clientes.

El presente es un estudio de las principales características de las plataformas para el desarrollo de soluciones de reportes, dado que, como se asienta en la hipótesis de este trabajo, de dichas características depende la capacidad para la definición y creación de reportes por parte de los desarrolladores de sistemas.

Actualmente no basta con que una herramienta funcione, se necesita que lo haga especialmente bien y esto se define con el cumplimiento de aspectos como cuán fácil o difícil es su uso o que tipo de usuario está en capacidad de aprovechar el sistema, eficiencia, flexibilidad, expansión, estabilidad, entre otros.

En el caso de plataformas para soluciones de informes, los requerimientos son especialmente complejos, debido a que no solamente se toma en cuenta los beneficios para el usuario final, sino también los beneficios para el desarrollador de informes. Y la fusión entre estos dos roles,

en el desarrollo de reportes AdHoc, donde el usuario final se convierte en el creador de reportes acorde a sus necesidades.

A lo largo de este trabajo, se presenta un análisis comparativo entre diversas herramientas de desarrollo de informes, así como una breve introducción a las tecnologías 2008 de Visual Studio .Net y SQL Server 2008, plataformas sobre las cuales se ha desarrollado la herramienta Dynamic Report Generator.



## **CAPITULO I: MARCO PROPOSITIVO**

### **1.1. ANTECEDENTES**

En la actualidad, toda entidad comercial necesita mantener una comprensión detallada sobre el funcionamiento de los negocios, esto exige soluciones que aprovechen el poder de los repositorios de datos ya existentes y que faciliten a los responsables de las decisiones la obtención de información oportuna, orientada a la acción, en forma de una alerta a través de un mensaje corto de correo electrónico, un informe altamente editado integrado en un portal o una herramienta analítica interactiva. La disponibilidad de esta información clave hace funcionar los negocios con éxito.

Uno de los retos más significativos a los que juntamente se enfrentan tanto la tecnología de la información como los negocios, es cómo sacar el máximo provecho de los activos de datos existentes para comprender y mejorar el desempeño comercial. Esto requiere dar la información adecuada a la persona adecuada con el nivel apropiado de interactividad.

Más que buscar una única solución a este problema para toda la empresa, muchas compañías han improvisado diversidad de tecnologías de Business Intelligence en un departamento o bien al nivel de un grupo de trabajo, para cubrir las necesidades de un proyecto concreto o un grupo de usuarios.

El resultado es todo un despliegue fragmentado de herramientas y sistemas que son costosos, pesados y difíciles de manejar y mantener, y a menudo demasiado especializados o intensivos en cuanto a recursos para servir a un amplio grupo de usuarios.

Servicios especializados manejan la generación de reportes o acceso de datos bajo demanda o bien planificación de informes basada en acontecimientos y tiempo.

Se conoce como Business Intelligence a las tecnologías, aplicaciones y prácticas para el almacenamiento, integración, análisis y presentación de información, su propósito es dar soporte para la toma de mejores decisiones empresariales, una de las sub ramas de Business Intelligence es "User/End-user Query and Reporting".

La presentación de informes constituye uno de los principales requerimientos en sistema informático, un informe es la materialización del estado de una entidad en un espacio y tiempo delimitado, y se hace indispensable para el análisis de diferentes ámbitos empresariales, tales como la economía, recursos humanos, actividades comerciales, entre otros.

En Seriva S.A., empresa orientada al desarrollo de software bancario, se conoce a fondo la importancia de los reportes en un sistema informático, durante varios años se ha invertido una gran cantidad de recursos en el desarrollo de informes y aplicaciones a medida para proveer importante información a los usuarios de los sistemas bancarios sobre su determinada área de trabajo.

Durante los últimos años Seriva forma parte de un gran proyecto, la "Suite Mercado de Capitales" o Suite MDC, misma que es un conjunto de aplicaciones independientes encargadas de automatizar los procesos mercantiles del mundo de la inversión de capital, cada una especializada en una actividad principal tales como, Inversiones en bolsa, acciones, bonos, prestación de garantías, el mercado cambiario, la custodia de bienes, entre otros. La Suite MDC en conjunto, constituye un gigantesco entorno mercantil invaluable para una institución financiera.

En el mercado bursátil las operaciones sobre acciones y bonos mueven una gran cantidad de dinero, y eso supone la existencia de grandes riesgos entre las partes involucradas, dado que las inversiones pueden resultar en pérdidas o ganancias.

Es aquí en donde se hace imprescindible la existencia de una fuente efectiva de información, que permita la toma de decisiones oportunas que aseguren la buena marcha de los negocios.

Sin embargo, durante las fases del proyecto Suite MDC, previas a la salida a producción, las definiciones de los usuarios han sufrido grandes cambios, varios informes solo conservan leves rangos de su especificación inicial. Y cada modificación se traduce en una inversión infructuosa de recursos humanos y económicos.

## **1.2. JUSTIFICACION**

La información es un recurso que es preciso gestionar de manera efectiva, al igual que los recursos financieros y humanos. Una gestión eficaz de los recursos de información constituye una condición básica para una buena gestión empresarial.

La velocidad a la que se requiere tomar decisiones en el mundo de los negocios de hoy, coloca unos requisitos adicionales al diseño de sistemas de información que permitan al usuario final en aprovecharlos de mejor manera.

Muchas son las herramientas que se han lanzado al mercado con el fin de satisfacer la necesidad de elaboración de informes, sin embargo, cada una de ellas será apropiada en función de varios factores del entorno en donde se la vaya a aplicar.

Actualmente, Business Objects Crystal Reports / Crystal Enterprise lidera el mercado de creación y distribución de informes, sin embargo Microsoft SQL Server Reporting Services, está ganando terreno rápidamente, tanto así que varias organizaciones orientadas al desarrollo de software están migrando sus módulos de reportes a Reporting Services.

Estas herramientas son poderosas plataformas para la creación de reportes, sin embargo, cada día se lanzan nuevos productos para dicho fin.

El presente estudio tiene como objeto principal la realización de un análisis comparativo de las principales plataformas para la elaboración de informes, de manera que se facilite la decisión de que tecnología utilizar para un caso en particular.

Se analizarán características decisivas en la construcción de una solución de reportes como:

- Opciones de conectividad y acceso eficiente a la información.
- Definición y edición de reportes. La facilidad y características disponibles para la creación de informes. Este es uno de los principales aspectos a considerar, debido a que esto determinará si se puede crear o no cierto tipo de reporte, y lo más importante, la capacidad de delegar la creación y edición de un reporte a los usuarios finales. El presente estudio se orientará específicamente a la capacidad de la plataforma para que la creación de reportes en tiempo de ejecución, mediante el desarrollo en un lenguaje de programación.
- El manejo de seguridades en el acceso a informes.
- Los medios y características de la distribución de informes.
- Escalabilidad y facilidad para adaptar la arquitectura a cada necesidad.

Como resultado final se presentará un cuadro de fortalezas y debilidades de cada plataforma así como un cuadro comparativo entre las características de las mismas. Información que constituirá un marco referencial para la toma de decisiones con respecto del uso de una determinada plataforma para la elaboración de informes.

Una vez que se tenga en claro las capacidades de cada tecnología, en el presente trabajo se seleccionará a una de ellas, para el desarrollo de la herramienta DYNAMIC REPORT GENERATOR, la misma que le permitirá al usuario final elaborar sus propios informes de una manera comprensible y adaptativa.

Dicha herramienta, se diseñará para integrarse a la perfección con datos existentes, facilitando la fiabilidad, escalabilidad y flexibilidad que el usuario necesita y espera de un sistema que constituya elemento básico de su infraestructura de tecnología de la información.

El resultado será una potente aplicación que ayudará a incrementar la productividad, la misma que podrá ser ensamblada en cualquier aplicación web de acceso a datos, mientras que reduce el costo de la elaboración de informes particulares, que normalmente se hacen vía especificación de requerimientos y una vez en producción, se requiere una importante inversión para eventuales cambios en los mismos.

La herramienta dynamic report generator formará parte de la Suite Mercado de Capitales de Seriva S.A., y estará disponible para el uso más conveniente que la EIS pueda destinar.

### **1.3. OBJETIVOS**

#### **1.3.1. Objetivo General**

Realizar un análisis comparativo de las principales plataformas para elaboración de informes en aplicaciones Web, haciendo uso de aquella que proporcione las mejores capacidades para la implementación de la herramienta Dynamic Report Generator de la Suite MDC de Seriva S.A.

#### **1.3.2. Objetivos Específicos**

- Recopilar información sobre plataformas para elaboración de informes identificando las principales utilizadas en el desarrollo informático.
- Analizar comparativamente las principales plataformas para elaboración de informes.
- Establecer las fortalezas y debilidades de las tecnologías estudiadas.
- Desarrollar un componente que sirva de medio para el acceso y manipulación de datos de cualquier base relacional, la misma que permitirá una interacción directa entre el usuario y los elementos de la base de datos.
- Implementar la herramienta utilitaria, DYNAMIC REPORT GENERATOR, que permitirá proveer al usuario final de la Suite MDC la capacidad de crear informes personalizados y administrarlos a conveniencia de una manera fácil e intuitiva.

### **1.4. HIPÓTESIS**

LAS CARACTERISTICAS DE LA PLATAFORMA UTILIZADA PARA LA ELABORACIÓN DE INFORMES EN UNA APLICACIÓN WEB INCIDEN DIRECTAMENTE SOBRE LA CAPACIDAD DE DEFINICIÓN Y CREACIÓN DE REPORTES.

## **CAPITULO II: MARCO TEÓRICO**

### **2.1. Introducción**

Los reportes de datos empresariales permiten identificar importantes mediciones y cifras de la situación económica, rendimiento, crecimiento empresarial, cumplimiento de metas y balances en general. Información que en una compañía resulta imprescindible a la hora de tomar decisiones.

Muchas son las tecnologías que se han lanzado al mercado con el fin de satisfacer las necesidades de creación y administración de informes. El propósito del presente estudio es identificar las principales herramientas de reporte del mercado, y entre ellas, determinar las características finales que podrán contener los informes desarrollados con las mismas.

Entre las principales características buscadas en las tecnologías orientadas a proveer servicios de creación y administración de informes están:

- Brindar una imagen clara del rendimiento de una organización
- Distribuir los informes en una amplia variedad de formatos.
- Distribuir los informes a demanda o vía suscripción en diferentes medios.
- Flexibilidad para reportes personalizados “fast ad-hoc reports”

## **2.2. Plataforma para elaboración de informes**

“En informática, una plataforma de desarrollo es el entorno de software común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada directamente a un sistema operativo; sin embargo, también es posible encontrarla ligada a una familia de lenguajes de programación o a una Interfaz de Programación de Aplicaciones (API).” (Wikipedia)<sup>1</sup>

Toda aquella funcionalidad, arquitectura, diseñadores, herramientas, asistentes, etc., que una plataforma para la elaboración de informes ofrece como parte de un producto distribuible, es la base fundamental para establecer la idoneidad, competitividad, destrezas y habilidades que tanto el desarrollador de informes como los usuarios finales, disponen para definir estructuras de reportes o crear informes acorde con los requerimientos explícitos e implícitos de un sistema informático.

Con propósitos de abreviación, en el presente trabajo se hará referencia a “Plataforma (s) para la Elaboración de Informes” mediante el acrónimo PEI.

## **2.3. Principales características de una PEI**

Cada una de las herramientas para elaboración de informes que han surgido, tiene como fin común cumplir con un conjunto de características que satisfagan las necesidades del área del mercado a quienes está dirigida. Evidentemente la meta de toda PEI, es abarcar la mayor cantidad de nichos en el mercado, por ello, la versatilidad y diversidad que ofrezcan es un factor determinante para captar la atención de los clientes.

Quizá la palabra informe, no proporciona una visión suficientemente amplia para considerar todos los elementos que giran alrededor de un reporte. Los mismos que en una aplicación de software se hacen imprescindibles a la hora de obtener estados del negocio y administrar la información siempre valiosa de un repositorio de datos. Es por ello que a continuación se presentan algunos de los conceptos asociados con una plataforma de elaboración de informes.

---

<sup>1</sup> Enciclopedia libre Wikipedia

### **2.3.1. Tipos de reportes**

Al hablar se tipos de reportes, se está haciendo referencia a la terminología utilizada para describir los diferentes tipos de informes y manera que pueden ser creados y utilizados. Un único reporte puede tener características de más de un tipo, por ejemplo, puede parametrizarse reportes snapshot, reportes ad-hoc incorporados mediante modelos en los que se basan y los subreportes pueden ser reportes enlazados.<sup>1</sup>

- Reportes Paramétricos
- Reportes Enlazados
- Reportes de Instantáneas
- Reportes Almacenados en Cache
- Reportes Ad hoc
- Reportes Clickthrough
- Reportes Drillthrough
- Subreports

#### **2.3.1.1. Reportes Paramétricos**

Un reporte paramétrico utiliza valores de entrada para el procesamiento de datos. Con un reporte parametrizado se puede variar la salida de los reportes basados en valores que son colocados cuando se ejecuta el reporte. Los reportes paramétricos son usados frecuentemente para reportes Drillthrough reportes, y subreportes conectando y filtrando reportes con datos relacionales.

#### **Parámetros**

Para seleccionar los datos a reportar, para filtrar los resultados que la consulta retorna, o para colocar propiedades de presentación o visualización de partes de un reporte.

---

<sup>1</sup> <http://msdn.microsoft.com/en-us/library/bb630404.aspx> (MSDN Microsoft)



También existen parámetros en cascada que populan una serie de dependencias, lista de parámetros, dropdown.

Los parámetros pueden servir para establecer vínculos entre reportes. Se puede ocultar parámetros en tiempo de ejecución.

Existen parámetros de consulta y parámetros de reporte, los parámetros de reporte difieren de los parámetros de consulta en el hecho que están definidos y procesados en un reporte, mediante el report server. Mientras que los parámetros de consulta están definidos como parte de una consulta a un conjunto de datos y son procesados por el servidor de base de datos.

#### **2.3.1.2. Reportes Enlazados.**

Un reporte enlazado es un ítem que provee un punto de acceso a un reporte existente. Conceptualmente esto es similar a un acceso directo a un programa.

#### **2.3.1.3. Reportes de Instantáneas**

Un snapshot de un reporte, contiene todo el esquema de información y los resultados de la consulta que fue ejecutada en un instante de tiempo específico. A diferencia de un reporte bajo demanda, en el cual los resultados se obtienen cuando se selecciona el reporte.

Los snapshots reports sirven para tres propósitos:

**Historia de Reportes.** Creando una serie de instantáneas se puede construir una historia de reportes que muestran como han cambiado los datos en la línea de tiempo.

**Consistencia.** Las comparaciones se realizan en instantes determinados de tiempo, por lo que se pierde las discrepancias que pueden surgir con múltiples usuarios que ejecutan los reportes en diferentes horas.

**Rendimiento.** Con la programación de reportes extensos durante horas de baja concurrencia, se puede reducir el impacto de procesamiento durante las horas habituales de trabajo.

#### **2.3.1.4. Reportes Almacenados en Cache**

Un reporte en cache, es una copia de un reporte procesado almacenada. Son utilizados para mejorar el rendimiento reduciendo el número de solicitudes del procesador de reportes y reduciendo el tiempo para recuperar reportes extensos. Estos tienen un tiempo mandatorio de expiración que normalmente se encuentran en minutos.

#### **2.3.1.5. Reportes Ad Hoc**

Los reportes Ad Hoc pueden basarse en modelos de datos o plantillas que presentan los datos del negocio en términos familiares para el usuario final, de modo que éstos pueden realizar sus consultas bajo demanda mediante una especie de autoservicio.

#### **2.3.1.6. Reportes Clickthrough**

Un reporte clickthrough es un reporte que despliega los datos relacionados desde un modelo de reporte cuando se da clic en datos interactivos contenidos dentro del modelo de reportes. La persona que creó el modelo determina que campos son interactivos y que campos son retornados cuando un reporte clickthrough se abre.

#### **2.3.1.7. Reportes Drillthrough**

Los reportes Drillthrough, son estándares que son accedidos a través de un hyperlink en un cuadro de texto en el reporte original, pueden ser filtrados por parámetros pero no necesariamente. Se diferencian de los subreportes en que estos no se despliegan con el reporte original, pero se abren separadamente. La diferencia con los reportes clickthrough, es que no se autogeneran desde una fuente de datos, en lugar de ello, son reportes personalizados que se almacenan en el report server.

#### **2.3.1.8. Subreportes**

Un sub reporte es un reporte que se muestra dentro del cuerpo de un reporte principal, conceptualmente un subreporte es similar a un frame en una página web. Es utilizado para embeber un reporte dentro de otro. Cualquier reporte puede utilizar como

subreporte de otro. Se puede configurar al reporte anfitrión para que pase parámetros al subreporte.

### **2.3.2. Madurez del Producto**

La experiencia en el mercado del desarrollo de soluciones para elaboración a informes, marca en gran medida la presencia en el mismo. El versionamiento de los productos hace efectiva la incorporación de nuevas características y corrección de funcionalidad errónea. Sin embargo, este no es un factor muy preponderante, debido a que muchas de las herramientas para elaboración de informes que no son conocidas, pueden proveer características superiores a los líderes del mercado.

### **2.3.3. Diseñador de informes**

El diseñador de informes es un entorno de desarrollo para el diseño y la construcción de informes. De sus características dependerá el esfuerzo de diseño que se deba aplicar, así como el aprovechamiento de la funcionalidad disponible.

### **2.3.4. Licenciamiento**

La distribución de los productos, derechos de uso de las herramientas y formas de licenciamiento son elementos que deben ser considerados en función del alcance la solución, dado que este es un tema que afecta directamente el presupuesto de los proyectos de desarrollo.

### **2.3.5. Formato del Archivo de Reporte**

Cada plataforma de elaboración de informes utiliza formatos propios para los archivos del reporte, así como diversos medios de almacenamiento. Es necesario tomar en cuenta este punto, debido a que en desarrollo de software es elemental el manejo de versionamiento, y es preciso conocer las posibles restricciones que surgirán en los manejadores de código.

### **2.3.6. Acceso a Datos**

El soporte que la plataforma de desarrollo de informes brinda para establecer la conectividad con las diversas fuentes de datos de la aplicación. Este es un tema importante si la empresa de desarrollo tiene aplicaciones que acceden a bases de datos heterogéneas o variedad de productos.

### **2.3.7. Data Binding**

Data binding es el proceso para obtener los datos desde una fuente dada y atarla a una interfaz. Comprende desde la carga de datos en un informe plano, hasta el manejo de carga en reportes sofisticados que contienen subreportes.

### **2.3.8. Características de Diseño de Reportes**

Este ítem abarca todas las características gráficas, de manipulación de objetos y campos que ofrece el diseñador de informes y determina el resultado que se puede obtener en la delineación de un informe.

- Capacidad WYSIWYG. Establece la precisión del informe resultante con respecto a la vista en tiempo de diseño, en muchas herramientas, la forma en la que se mira el informe en diseño discrepa totalmente con la vista previa e impresión.
- Estructura del Reporte. Como se administran las regiones o secciones de un informe. En varias herramientas, la distribución de regiones imposibilita la aplicación de funcionalidades como agrupación.
- Fórmulas y Cálculos. Contempla el repositorio de funciones y operadores que la herramienta ofrece para el establecimiento expresiones matemáticas, lógicas, textuales, etc.
- Agrupación/Subtotales. Los factores a tomar en cuenta con respecto a la agrupación y subtotales, básicamente se resumen en anidamiento, y soporte para regiones independientes.

- Extensibilidad. Soporte para inclusión de código fuente externo al diseñador de informes.
- Secciones para administración de flujo de datos. Manejo de saltos de secciones o páginas.
- Despliegue del Reporte en una Página Web. Fidelidad de la presentación con respecto a la vista previa del diseñador.
- Texto Enriquecido. Soporte para incluir campos con texto enriquecido.

### **2.3.9. Formatos de Exportación**

A cuántos y que formatos es posible exportar un reporte generado, este es un punto débil en muchas tecnologías para elaboración de informes, lo ideal es el soporte de los formatos más comunes como HTML, PDF, Excel, Word, Imagen.

### **2.3.10. Manejo de Parámetros**

La gran mayoría de aplicaciones, requieren que los informes sean paramétricos, adicionalmente es muy solicitado que los valores disponibles para los parámetros sean obtenidos y delimitados por la información de las propias bases de datos de la aplicación, aquí nuevamente es importante evaluar términos de anidamiento.

### **2.3.11. Arquitectura**

La arquitectura es un aspecto determinante en temas de escalabilidad y administración que el cliente requiere, las plataformas de elaboración de informes deben ajustarse a todas las necesidades.

**Soporte multi capa.** Una arquitectura en capas permite una distribución del proceso basado en la funcionalidad.

**Cliente.** Herramientas para la interacción con el sistema de usuarios y administradores.

**Servidor.** Capa de administración de procesos y administración de módulos, manejo de solicitudes, programación y distribución

**Procesamiento de reportes.** Capa para acceder a las fuentes de datos y reportes.

### **2.3.12. Costos**

Aunque hablar de temas económicos no está relacionado directamente con las prestaciones que una plataforma para desarrollo de reportes pueda tener. No es menos importante, debido a que es necesario tomar en cuenta que algunas herramientas, requieren una licencia por cada usuario cliente de la aplicación, lo cual sin duda impacta en el presupuesto de un proyecto.

### **2.3.13. Reportes AdHoc**

La tendencia actual, es brindar las facilidades a los usuarios finales, para que satisfagan por si mismo las necesidades particulares de información. Muchas herramientas promocionan un soporte AdHoc completo, sin embargo la realidad es que siempre habrá limitaciones como el pre diseño de plantillas o modelos de datos.

### **2.3.14. Gráficas**

Dependiendo del área de aplicación de un informe, normalmente en reportes gerenciales, se requiere la presentación de gráficas estadísticas o medidores en una amplia gama de modelos, una plataforma para elaboración de informes no puede estar completa sin tener un buen soporte gráfico.

### **2.3.15. Rendimiento**

El tiempo de renderización de un informe también constituye un elemento clave, las buenas prácticas para desarrollo de informes sugieren que un reporte no debe exceder los 30 segundos de espera por parte del usuario, ya que lo contrario puede causar frustración. Sin embargo, el tiempo de presentación no está determinado únicamente por el número de datos ni la optimización de las consultas. La arquitectura y conectividad juegan un rol decisivo en los tiempos de respuesta.

### **2.3.16. Escalabilidad**

La facilidad para adaptar la arquitectura a cada necesidad, modularidad del producto, es deseable un diseño que escale fácilmente para soportar entornos de alto volumen. Que exista la capacidad de crear una granja de servidores para reporting, con múltiples servidores de reportes, accediendo en el mismo core de reportes, sirviendo a una vasta cantidad de usuarios de forma simultánea.

### **2.3.17. Seguridad**

Es elemental pensar que la presentación de datos es de vital reserva para una compañía, el control de autenticación para el acceso de informes se lleva a cabo de diferentes formas en las herramientas para elaboración de informes. En este aspecto es importante evaluar elementos como encriptación, consola de administración, seguridad integrada, autenticación programática, manejo de roles, entre otros.

## **2.4. Capacidad de definición y creación de reportes**

### **2.4.1. Capacidad para publicación de reportes**

Los mecanismos utilizados para publicar un informe dependen en gran medida de la arquitectura de la plataforma, es fundamental considerar la facilidad de colocar una solución de reportes en el ambiente de producción. Así como los medios adicionales disponibles para entrega como suscripciones, programación, correo electrónico, etc.

## **2.5. Principales PEI del mercado**

Se conoce como Herramientas de Reporting al Software diseñado a generar reportes basados en una amplia variedad de fuentes para permitir a las organizaciones entender de mejor manera sus empresas.

Los mejores en el campo de herramientas de reporting incluyen a Cognos, Actuate, SAP Business Objects, Microsoft, Oracle e Hyperion<sup>1</sup>. Cada uno de ellos provee paquetes completos

---

<sup>1</sup> (Dashboard Spy, 2007) Una de las principales fuentes de las que se partió, para determinar las tecnologías a analizar.

de BI, de los cuales reporting es una función integral, generalmente, en los paquetes de reporting, los usuarios juzgan lo siguiente:

- Facilidad para usar la interfaz.
- Características de personalización.
- Formatos y plataformas para exportar.
- Acceso a datos e integración con otras aplicaciones.
- Formato WYSIWYG.
- Habilidad para crear reportes adhoc.

Como parte del análisis inicial, se citaran las características de algunas de las principales aplicaciones proveedoras de servicios para reporting. Para establecer una cuantificación de dichas características se utilizará la siguiente tabla que asocia una valoración. A ésta valoración se añade un ajuste de 1 punto a cada parámetro utilizado con el fin de incorporar un elemento de cuantificación medidora adicional a la existencia de una caraterística.

**Tabla II-1.** Parámetros generales para la valoración de una PEI

	Parámetro	Código	Puntaje ± 1	Valor
<b>Facilidad de Uso</b>	Tiene asistentes disponibles	FU1	5	10
	Distribución de secciones intuitiva	FU2	3	
	Interfaz gráfica familiar para el desarrollador	FU3	2	
<b>Personalización</b>	Fórmulas	P1	2.5	10
	Formato condicional	P4	2.5	
	Subreportes	P3	2.5	
	Parámetros en cascada	P4	2.5	
<b>Formatos de Exportación</b>	Formatos HTML, PDF, Excel, Word, Imagen	FE1	5	10
	Mantiene todos los elementos en el formato exportado	FE2	3	
	Exportación de gráficos y subreportes	FE3	2	
<b>WYSIWYG</b>	Mantiene la ubicación y características de los controles	W1	5	10
	Mantiene la presentación tanto en aplicaciones Windows como aplicaciones web	W2	3	
	Evita la aglomeración de separadores en tiempo de diseño	W3	2	
<b>Reportes Ad Hoc</b>	Soporta definición de reportes por parte del usuario final.	AH1	6	10
	Independencia de modelos o plantillas previas	AH2	1	
	Capacidad para predefinir el acceso a los datos sin intervención del usuario	AH3	3	



Entre las principales tecnologías que abarcan el mercado del desarrollo de soluciones para presentación de informes se encuentran:

- Crystal Reports
- SQL Server Reporting Services
- Performance Point Server 2007
- SQL 2008 Report Builder
- Access Reports
- Excel 2007 Connecting to Olap Cubes & Relational Databases
- XSL FO
- Report Sharp –Shooter
- BinaryWorX BXQuery – ActiveX
- SBS Development
  - XL-DBQuery
  - Automated-SQL-Builder – ActiveX
- TrendAnalyzer
- Total Access Memo – ActiveX
- Aspanse AdHoc - .Net
- ComponentOne Studio
- DXperience
- Data Dynamics
  - ActiveReports for .Net
  - ActiveReports
  - DynamiCube
  - ContourCube
- Clover

Según el criterio de evaluación de Forrester (Evelson, 2008), los vendedores que se encuentran liderando el mercado son IBM Cognos y SAP Business Objects, mientras que Oracle y SAS Institute se acercan a posiciones de liderazgo en BI empresarial gracias a la riqueza de su funcionalidad, facilidad para escalar y la integridad de su corporación así

como la visión y estrategia de los productos. Actuate, Information Builders, Microsoft, MicroStrategy, SAP, y productos nuevos como TIBCO Spotfire, entran con fortaleza siguiendo muy de cerca, prácticamente en los talones de los líderes, ofreciendo alternativas muy respetables y una multitud de elecciones para la administración y conocimiento de la información (I&KM). Otras plataformas como Panorama Software y QlikTech, que carecen de características que lo refuten como simples, son también contendores respetables, eventualmente pueden funcionar mejor que los líderes.

A continuación se nombra algunas de las características principales de las herramientas nombradas.

### **2.5.1. SAP Business Objects Crystal Reports 2008**



Proviene de Business Objects, recientemente adquirido por SAP.

Aplicación que provee servicios de presentación de informes a los usuarios, está diseñada para crear reportes flexibles y con alta fidelidad e integrarlos tanto a aplicaciones Windows como a aplicaciones Web. Crystal Reports para Visual Studio .Net, provee una solución integral para los desarrolladores de .Net, que está completamente integrado con el IDE de Visual Studio .Net y el .Net Framework. Crystal Reports soporta ADO.NET, XAML, Web Services y los controles ASP.NET y almacenamiento en caché. También se integra perfectamente con el Server Explorer de Visual Studio .Net, el toolbox y el ambiente de diseño. Tiene un modelo enriquecido de programación y opciones flexibles para la personalización y distribución de informes.

Las últimas versiones de Crystal Reports tienen características mejoradas de configuraciones de exportación incluyendo formatos RTF, HTML, un workbench y funciones de ordenamiento mejoradas, Report prompts que ayudan a los usuarios a interactuar con cada ejecución. Las definiciones ahora pueden ser compartidas entre múltiples reportes, simplificando el proceso de diseño. La clasificación en grupos de valores puede tener sus propias fórmulas, que ordenan los reportes consolidándolos vía un mecanismo de parámetros. Las fórmulas de ordenamiento usadas en los parámetros permiten a los

usuarios con diferentes necesidades de ordenamiento acceder a un mismo reporte. Esta característica reduce el número de reportes a los que se requiere dar mantenimiento.

Las características de Crystal Reports 'HTML y presentación WYSIWYG, agilizan el tiempo de desarrollo y eliminan la necesidad de publicar los reportes cada vez que son modificados en tiempo de diseño. Hay un método automático para añadir datos de reportes dentro de gráficas estadísticas y a través de matrices sin necesidad de crear ningún link.

Las gráficas y matrices pueden ser generadas automáticamente durante el proceso de diseño. A través de un controlador nativo de XML, se puede convertir archivos XML puros en reportes con formato e incluso transformar XML en otros formatos como HTML y WML. El controlador acepta tanto DTD (Document Type Definition) y schemas, trabaja con un top de Java's J2SE 1.4 SDK. El software también trae un DataDirect que acepta XML sobre fuentes de datos ODBC. Este acceso está limitado a estructuras específicas pero pueden aceptar múltiples tipos de archivos como "data island"<sup>1</sup>, ADO (ActiveX Data Objects) y archivos con formatos jerárquicos.

Crystal Reports' OLAP, experto en diseño de reportes, provee seis pasos para procesamientos de cubos OLAP. El software provee conectividad OLAP hacia IBM's y servidores OLAP DB2, Hyperion Essbase, Holos HDC Cube y servicios OLAP de Microsoft. Adicionalmente a la conectividad OLAP, Crystal Reports provee acceso directo a cubos OLAP sobre archivos locales y distribuidos como cubos HTTP. Para permitir acceso a los archivos de cubos, Crystal Reports separa las dimensiones OLAP en tres estructuras, permitiendo a los diseñadores de reportes trabajar con datos OLAP de la misma manera que con datos relacionales. Sin la función de repositorio, los desarrolladores pueden añadir funciones Java personalizadas. Esto significa que cualquier dato accedido desde aplicaciones Java usando Enterprise JavaBeans o colas de mensajes Java puede ser usado con Crystal Reports. Las características Java Server están disponibles a través de un nuevo JSF developer kit. Crystal Reports proporciona un control de programación completo a través de su API.

---

<sup>1</sup> XML data insland son datos XML embebidos en una página HTML

### **Principales Características**

- Acceso a Datos: Accede a cualquier clase de datos con más de 35 data drivers, acceso a datos personalizados, y un control severo sobre conectividad de bases.
- La información específica puede ser resaltada para los usuarios finales.
- Proporciona al usuario la opción de pre visualizar los reportes antes de publicarlos sobre la web.
- Drill down sobre datos relacionales o datos OLAP para relevar los detalles que pueden estar pasando desapercibidos.
- Formateo y diseño: Puede ayudar a obtener un diseño profesional, reportes interactivos usando características como el diseñador visual de reportes e indicaciones dinámicas.
- Visualización de reportes e interacción: Los usuarios finales pueden ver y trabajar con sus propios reportes a través de la personalización para ajustarlos a sus necesidades.

### **Limitaciones**

Crystal Reports ha sido un generador de reportes popular, pero debido a la inherente deficiencia de su diseño arquitectónico, Crystal Reports tiene algunas limitaciones incluyendo:

- Incapacidad de modificar los reportes generados.- Una vez que se usa Crystal Reports para generar los reportes en runtime, no se tiene la opción de permitir a los usuarios editar y/o modificar los reportes.
- Incapacidad de dibujo técnico.- no se puede crear e incluir dibujos técnicos sofisticados en los reportes basados en texto creados por Crystal Reports.
- No tiene soporte de profundidad en sub reportes, solo soporta 1 nivel.
- No tiene soporte para acceso a datos en tiempo real o capacidades EII

Crystal Reports ha sido la solución de presentación de informes premier durante muchos años, y sólo ahora está comenzando a ser sustituida por Microsoft Reporting Services.

**Tabla II-2.** Valoración General Crystal Reports

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			9
	Puntaje ± 1	5	2.5	1.5			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		8
	Puntaje ± 1	2	2	2	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			9
	Puntaje ± 1	5	2.5	1.5			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			9
	Puntaje ± 1	5	2.5	1.5			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			8
	Puntaje ± 1	5.5	0	2.5			

### 2.5.2. SAS® Web Report Studio

Consultas basadas en Web y elaboración de informes diseñado especialmente para usuario de negocio. SAS Web Report Studio está diseñado para satisfacer las necesidades específicas y niveles de destreza de usuarios no técnicos en la organización.

#### Características Principales

- Fortalece la capacidad de los usuarios para un autoservicio en consultas y elaboración de informes. Los datos se presentan en términos del negocio.
- Mejora el valor empresarial de IT y su administración de datos, se elimina el riesgo de resultados inconsistentes, y los administradores de la empresa obtienen información confiable en la que pueden creer.
- Minimiza los costos de adiestramiento y soporte, dado que la interfaz de usuario es tan simple que los usuarios novatos pueden satisfacer sus necesidades rápidamente.
- Provee tabla de contenidos, y drill y expand para tablas y gráficas.
- Ejecuta interacciones OLAP específicas.
- Autorización y distribución basado en roles.
- Arquitectura abierta basada en el framework estándar J2EE.
- Soporta una gran variedad de bases de datos relacionales así como OLAP. Los reportes pueden tener múltiples fuentes de datos.

#### Limitaciones

- No soporta dependencia entre parámetros.

- Actualmente se necesita utilizar el Internet Explorer versión 5 (5.5+) o versión 6. Si se ha actualizado a IE 7 se debe hacer un rollback a IE6.
- Actualmente el usuario final únicamente puede exportar a Excel.

**Tabla II-3-** Valoración General SAS Web Report Studio

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			9
	Puntaje ± 1	5	2.5	1.5			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		8
	Puntaje ± 1	2	2	2	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			7
	Puntaje ± 1	4.5	2.5	0			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			7
	Puntaje ± 1	4.5	2.5	0			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			8
	Puntaje ± 1	5.5	0	2.5			

### 2.5.3. SQL Server Reporting Services 2008



Microsoft Reporting Services proporciona un enfoque integrado, un conjunto de herramientas end-to-end, para creación, administración y entrega de informes. Incluye un motor escalable, basado en servidor para hosting y procesamiento de informes.

Los informes se distribuyen en formatos HTML / PDF / Excel / CSV / WORD, desde un servidor central, o integrado en aplicaciones Windows. La entrega de informes puede ser programada, pudiendo ser enviada vía e-mail, o mediante repositorios compartidos. Reporting Services viene gratis con la licencia de SQL Server.

La versión introducida en SQL 2005 contiene varias mejoras sobre la primera versión SQL 2000, como parámetros multivalor, un report manager especializado, entre otras. SQL 2008 Reporting Services se presenta con mayores fortalezas, como la radicalización de reportes adhoc mediante la generación de reports models previos a ser utilizados por la herramienta relacionada Report Builder 2.0.

SQL Server Reporting Service (SSRS), es un grupo de aplicaciones interrelacionadas que interactúan con el motor de base de datos SQL Server. SSRS incluye todas las piezas para el desarrollo y administración para publicar los informes del usuario final en diversos formatos.

Si se tiene experiencia con herramientas como Crystal Reports, o Microsoft Access, entonces SQL Server Reporting Services resultará bastante familiar. Incluye un constructor de reportes drag and drop llamado Report Designer. Adicionalmente, SSRS también incluye una aplicación para permitir a los usuarios finales crear sus propios reportes ad hoc llamado Report Builder. Una característica única de SSRS es la capacidad para crear suscripciones a los reportes, donde los usuarios pueden solicitar los reportes para ser enviados por e-mail automáticamente en varios intervalos.

Otra característica es el Report Manager, que es una aplicación web para la organización, seguridad y despliegue de los reportes creados, el report manager ejecuta las siguientes funciones:

- Visualización de Reportes.
- Búsqueda y navegación por reportes específicos.
- Configuración de seguridad en la visualización de reportes.
- Creación de programación y suscripciones.
- Lanzamiento de Report Builder para reportes ad hoc.

#### **Principales Características**

- En acceso de datos soporta los drivers más comunes en el mercado, incluyendo ADO, XML, cubos OLAP, ODBC, entre otros.
- SSRS puede ser utilizado a través de la organización por varios usuarios simultáneamente.
- SSRS aprovecha las ventajas de .Net y de sus proveedores de datos, puede consultar una variedad de fuentes de datos como SQL Server, Oracle, Analysis Services, Microsoft Access entre otros. Si es necesario, SSRS también puede aplicarse a través de las extensiones personalizables de procesamiento de datos.
- SSRS incluye mecanismos para la distribución de reportes a usuarios individuales bajo demanda o bajo una tarea programada.

- Los informes SSRS pueden ser subministrados en un número de formatos, incluyendo Excel o HTML. Los informes pueden ser utilizados por sí mismo como una fuente de datos para otras aplicaciones cuando son presentados en XML.
- SSRS proporciona mecanismos para presentación de informes ad hoc.
- Buena usabilidad y capacidades para desarrollo de aplicaciones, con una funcionalidad analítica promedio resaltada con la funcionalidad OLAP en Analysis Services.

### **Limitaciones**

- Una de las características bastante criticadas sobre SQL Server Reporting Services es el manejo de licencias, a pesar que mirando de una forma superficial, parezca atractivo el hecho de tenerlo gratis con la adquisición de la licencia de SQL Server. Si se analiza más a fondo, es un truco de mercadeo. Si se quiere instalar SSRS sobre un servidor standalone, se debe comprar una licencia adicional de SQL Server por cada computadora. La razón por la cual esto es importante es porque SSRS es un recurso muy intensivo. La presentación de informes consume más recursos de CPU y memoria y estos son los mismos recursos por los que SQL Server está compitiendo. Muchas empresas consideran que no es práctico para mantener SSRS en el mismo servidor como SQL Server.
- Las versiones previas a Reporting Services 2008 presentaban ciertas limitaciones del diseñador. Primero es que no hay un diseñador verdaderamente WYSIWYG. La forma en la que un reporte aparece en vista previa puede ser diferente a lo que se ve cuando el reporte es impreso. SSRS no es tan exacto como Crystal Reports porque está en una versión inicial y eso crea problemas con usuarios que requieren precisión para presentación de reportes. Esto sin embargo no es de mucha importancia para un usuario de negocios. Aunque SSRS proporciona sub informes a fondo, hasta 20 niveles, se tiene problemas al exportar sub-informes a Excel. La exportación a Excel solo admite listas de objetos anidados. SSRS tampoco es compatible con el formato dentro de diferentes objetos. Se tiene que crear varios objetos, cada uno con su propio




formato, y colocarlos uno al lado del otro. Estas limitaciones han sido cubiertas por la versión 2008.

**Tabla II-4. Valoración General SSRS**

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			9
	Puntaje ± 1	5	2.5	1.5			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		8.5
	Puntaje ± 1	2	2.5	2	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			8
	Puntaje ± 1	4	2	2			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			9
	Puntaje ± 1	4.5	2.5	2			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			9
	Puntaje ± 1	6	0.5	2.5			

#### 2.5.4. Performance Point Server 2007

Construido sobre SQL Server 2008 y distribuido a través de Office  SharePoint Server y Excel 2007, Microsoft PerformancePoint Server 2007 es una aplicación de Business Intelligence que provee información para supervisión, análisis y planificación.

El uso de la plataforma Microsoft Business Intelligence, PerformancePoint Server 2007 incorpora a una aplicación gestión de rendimiento escalable. En integración de datos, un modelo de análisis de datos que incluye un motor de cálculo y minería de datos, y una plataforma flexible y escalable de información para transmitirla hacia una amplia audiencia. Todo ello se entrega a través de la Web en Microsoft Office SharePoint Server 2007, ofreciendo a un lugar central para obtener la información de business intelligence.

#### Principales Características

- PerformancePoint Server tiene funcionalidad que permite a los usuarios finales crear sus propios reportes, dashboards y scorecards. Está integrado con Reporting Services y Share Point.

#### Limitaciones

- PerformancePoint requiere el soporte de herramientas externas como Reporting Services para proveer un servicio completo de presentación de informes, dado que su especialización es proveer herramientas de monitoreo y control.

**Tabla II-5.** Valoración General Performance Point Server

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			6
	Puntaje ± 1	4.5	0	1.5			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		5
	Puntaje ± 1	2.5	2.5	0	0		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			5
	Puntaje ± 1	5	0	0			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			7
	Puntaje ± 1	5	0	2			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			0
	Puntaje ± 1	0	0	0			

### 2.5.5. SQL 2008 Report Builder



SQL Report Builder es una herramienta gratuita que viene con SQL Server 2008, y permite a usuarios no técnicos crear fácilmente informes sencillos. La interfaz de usuario es intuitiva y familiar para cualquiera que ha creado informes de MS Access. Mantiene toda la esencia de creación de informes de Reporting Service. Está basado en la publicación previa de modelos de datos que facilitan la interpretación de los elementos de la base de datos, así como de templates solo los cuales los usuarios finales pueden incluir sus propias modificaciones

#### Características Principales

- Report Builder es únicamente un editor de informes, requiere de los servicios de reporting services para administrar y publicar los informes, se puede trabajar sin conexión para crear informes.
- El primer release de Report Builder 2.0, no soporta Custom Data Extension localmente. Es por ello que no es posible crear un reporte completo basado en los datos de extensión Enesys RS a partir de cero, sin embargo algo interesante de Report Builder es la capacidad de editar directamente un informe almacena en un servidor de informes.

### Limitaciones

- A pesar que es una de las principales herramientas que verdaderamente provee la capacidad para crear reportes ad hoc, la manipulación de datos por parte del usuario final requiere la creación de un modelo de reporte en ambiente de desarrollo.
- Report Builder 2.0 solo puede producir reportes compatibles con SSRS 2008.

**Tabla II-6.** Valoración General SQL 2008 Report Builder

		Valoración				
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>		9
	Puntaje ± 1	4.5	3	1.5		
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	9
	Puntaje ± 1	2	2	2.5	2.5	
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>		8
	Puntaje ± 1	4	2	2		
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		9
	Puntaje ± 1	4.5	2.5	2		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		9
	Puntaje ± 1	6	0	3		

### 2.5.6. Access Reports



Microsoft Access tiene características para creación de informes sencillos. Estos informes pueden ser manejados de un acceso o SQL Server backend. Generalmente se recomienda migrar este tipo de reportes a Reporting Services.

### Características Principales

- Presenta varias plantillas a partir de las cuales se puede incluir las variantes requeridas.
- Los reportes de Access pueden usar VBA, macros y formas para dar formato al reporte dinámicamente.

### Limitaciones

- A parte de las limitaciones en su repositorio de datos, los reportes en Access presentan límites de tamaños de páginas.
- Hay limitación en el número de sub formas y sub reportes anidados.

- Microsoft Jet solo puede manejar un número limitado de sesiones. Si una aplicación usa un gran número de controles de datos ADO, Jet puede quedarse sin recursos.
- Jet puede soportar hasta 255 usuarios concurrentes, pero el rendimiento de la arquitectura basada en archivos puede impedir el uso para muchos de los usuarios, se recomienda usar Jet para 10 usuarios o menos.

**Tabla II-7.** Valoración General Access Reports

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			6
	Puntaje ± 1	4.5	0	1.5			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		5
	Puntaje ± 1	0	0	2.5	2.5		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			5
	Puntaje ± 1	5	0	0			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			6
	Puntaje ± 1	4.5	0	1.5			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			0
	Puntaje ± 1	0	0	0			

### 2.5.7. Excel 2007 Connecting to Olap Cubes & Relational Databases

Excel 2007 provee características de gran alcance y herramientas fáciles de utilizar para crear informes de cubos OLAP y bases de datos relacionales. Esto incluye el filtrado slicing & dicing y clasificación.

Además, con SharePoint Services Excel 2007, se puede publicar hojas de cálculo de Excel en la Web, permitiendo que las personas no técnicas tales como BA y contadores creen y desplieguen informes significativos.

**Tabla II-8.** Valoración General Excel 2007 Connecting to Olap Cubes

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			6
	Puntaje ± 1	4.5	0	1.5			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		7
	Puntaje ± 1	2.5	2	0	2.5		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			0
	Puntaje ± 1	0	0	0			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			8
	Puntaje ± 1	5	3	0			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			8
	Puntaje ± 1	5.5	0	2.5			

### 2.5.8. XSL FO



XSL FO es un documento basado en el motor XML de datos y hojas de estilo XSL. Esto permite una ingeniería de documentos bastante flexible que es más apta en la producción de documentos expertos en lugar de informes. Esta es una limitación de algunas otras tecnologías de información, mencionadas. Varias tecnologías son muy buenas en la creación de informes, pero no tanto en creación de documentos.

Un documento XSL-FO es un documento XML en el que se especifica cómo se van a formatear unos datos para presentarlos en pantalla, papel u otros medios. El significado de las siglas XSL-FO es eXtensible Stylesheet Language Formatting Objects. Hay que destacar que en el documento XSL-FO figuran tanto los datos como el formato que se les va a aplicar.

Los procesadores XSL-FO libres más conocidos son FOP y PassiveTeX.

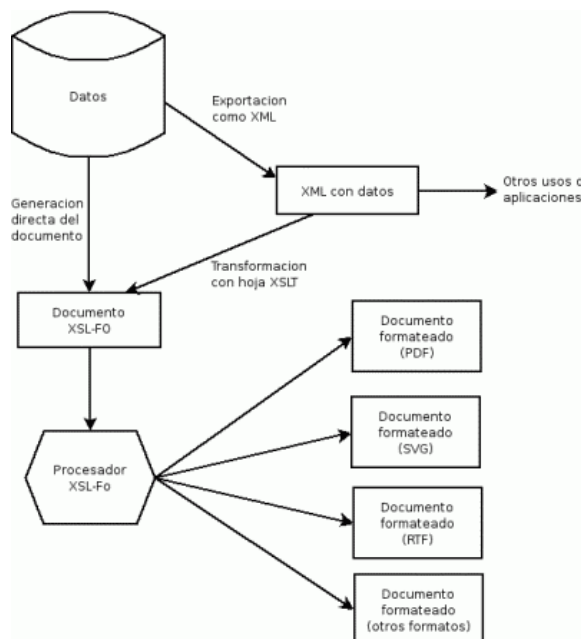


Figura I. Tratamiento de documentos en XSL - FO

#### Principales Características

- Principalmente se lo utiliza para implementar reportes en aplicaciones desarrolladas en Java.

- La unidad básica de trabajo en un documento XSL-FO es el "Formating Object", unidad básica para presentar (formatear) la información. Estos objetos de formato se refieren a páginas, párrafos, tablas, etc.
- La estructura de un documento FO tiene poderosas analogías con HTML, se tiene tablas, filas, celdas, entre otros elementos.
- FOP no es una solución para construcción de reportes standalone, es solo una manera de transformar archivos XSL-FO a un reporte, por ello para poblar los datos del reporte es necesario utilizar un motor adicional como Jakarta Velocity. La creación de un reporte FOP se da en dos fases, crear el reporte XML vía Velocity, alimentar la secuencia XML hacia FOP.
- Comparado con otras tecnologías como Jasper, FOP tiene bajo rendimiento debido a que almacena en cache el driver y el template de Velocity entre ejecuciones sucesivas.

### Limitaciones

- Entre las principales limitaciones de XSL-FO consta la dificultad de modificar el diseño de un reporte así como que sus extensiones no son reconocidas de forma transparente por aplicaciones relacionadas como DreamWeaver o editores HTML.
- Apache FOP ha presentado discontinuidad en la presentación de actualizaciones y nuevas versiones, el tiempo de respuesta y el consumo de recursos han sido aspectos bastante criticados.
- No existe suficiente documentación para el uso de esta tecnología y tampoco de los editores como FOP y PasiveText.

**Tabla II-9.** Valoración General XSL FO

		Valoración				
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>		7
	Puntaje ± 1	5	0	2		
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	7
	Puntaje ± 1	2.5	2.5	2	0	
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>		9
	Puntaje ± 1	5	2.5	1.5		
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		8
	Puntaje ± 1	5	3	0		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		0
	Puntaje ± 1	0	0	0		

### 2.5.9. Report Sharp –Shooter



Report Sharp –Shooter en sus dos versiones Web Forms Edition, y Lite Edition, provee una presentación de informes flexible y fácil de usar. Está diseñado para el .NET Framework y escrito en C#. Proporciona la capacidad para generar informes de cualquier complejidad, de múltiples fuentes de datos y exportar los informes generados en la mayoría de formatos, como PDF, HTML, EMF, BMP, JPG, GIF, TIFF, PNG, Excel, Excel (XML), CSV, TXT y RTF. También se pueden utilizar los filtros personalizados de exportación. Incluye un diseñador para el usuario final que puede ser usado tanto para plantillas de informes y documentos, también se puede ver los informes creados en Windows o aplicaciones Web utilizando los viewers suministrados.

#### Principales Características

- Report Sharp Shooter-Lite Edition ofrece productos de funcionalidad básica y está orientado a aplicaciones Windows. Es la solución ideal para los desarrolladores, cuyo proyecto no tiene que proporcionar al usuario la posibilidad de modificar los informes.
- Report Sharp-Shooter introduce un concepto avanzado de bandas SideBySide que permite colocar partes separadas del informe en paralelo con desplazamiento horizontal.
- Incluye una arquitectura plug-in abierta, mediante la cual los desarrolladores pueden añadir funcionalidad personalizada, así como crear filtros de exportación.

#### Limitaciones

- La versión express contiene limitaciones en el uso de componentes para gráficas y tablas pivot.
- Una vez que un informe es publicado, no existe forma que los usuarios finales puedan modificarlo.

**Tabla II-10.** Valoración General Report Sharp –Shooter

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			7
	Puntaje ± 1	4.5	2.5	0			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		8
	Puntaje ± 1	2	2	2	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			8
	Puntaje ± 1	4.5	2.5	1.5			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			7
	Puntaje ± 1	5	3	0			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			0
	Puntaje ± 1	0	0	0			

### 2.5.10. BinaryWorX BXQuery – ActiveX



BinaryWorX BXQuery ha sido diseñado para permitir a los usuarios finales extraer datos de sus aplicaciones ad hoc. Con una intuitiva interfaz de usuario, BXQuery hace que, para la tarea de extracción de datos ad hoc desde una aplicación, baste con añadir los elementos que le forman, y con unas pocas líneas de código, se obtenga la interfaz deseada. BXQuery es la última serie de componentes para la construcción de la consulta personalizada.

#### Características Principales

- BXQuery es nunca necesita conectarse a la fuente de datos actual, el diccionario de datos define toda la información del esquema (tablas, campos, etc.) necesarios para construir una consulta. Se puede definir un diccionario de datos propio o conectarse a la fuente de datos para permitir que BXQuery lo haga automáticamente.
- Soporta tablas derivadas (tablas basadas en sub consultas) en el diseñador de consultas.
- Se puede limitar las tablas y campos disponibles para el usuario final.
- La interfaz de usuario es similar a Microsoft Access o Microsoft SQL Server.

#### Limitaciones

- No existe suficiente documentación de la herramienta.



**Tabla II-11.** Valoración General BinaryWorX BXQuery

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			7
	Puntaje ± 1	4	2	1			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		7
	Puntaje ± 1	2.5	2.5	0	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			0
	Puntaje ± 1	0	0	0			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			7
	Puntaje ± 1	5	0	2			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			7
	Puntaje ± 1	5	0	2			

### 2.5.11. SBS Development

#### 2.5.11.1. XL-DBQuery



XL-DBQuery es una herramienta para análisis de datos y presentación de informes para Excel que enlaza el análisis y

capacidad de presentación de reportes de Excel con MS Query. Está basado en el popular Query Builder 'Automated-SQL-Builder'. Con poco esfuerzo los usuarios pueden conectar, consultar y analizar datos corporativos usando la funcionalidad familiar de Excel para crear reportes basados en los datos provenientes de XL-DBQuery.

#### Características Principales

- Es un add in para Microsoft Excel que permite consular y analizar datos.

#### Limitaciones

- Es una herramienta de reportes con elementos básicos, básicamente constituye una herramienta de minería de datos.
- No existe suficiente documentación.

**Tabla II-12.** Valoración General XL-DBQuery

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			6
	Puntaje ± 1	4	0	2			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		6
	Puntaje ± 1	2.5	2.5	0	1		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			5
	Puntaje ± 1	4	1	0			

<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		0
	Puntaje ± 1	0	0	0		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		7
	Puntaje ± 1	5	0	2		

### 2.5.11.2. Automated-SQL-Builder – ActiveX



Proporciona a los usuarios de aplicaciones cliente servidor y aplicaciones web la capacidad de crear y ejecutar sus propias consultas. Automated-SQL-Builder es un control ActiveX que puede ser usado para proveer a cualquier aplicación la capacidad de consultas de usuarios para las soluciones de reportes. El control contiene una herramienta que permite a los usuarios de todos los niveles técnicos crear y ejecutar sin problemas consultas ‘SELECT’ de SQL, de una forma muy intuitiva y natural.

#### Características Principales

- El modo de creación de consultas es muy similar al de Microsoft Access, y su uso es recomendado para áreas de consulta específicas que no necesitan mayor complejidad.
- Es compatible con .Net.
- Se puede añadir o remover tablas en runtime.
- Se puede trabajar tanto en modo bound como unbound.
- Soporta alias para tablas y campos, de forma que su interpretación sea más natural para el usuario final.

#### Limitaciones

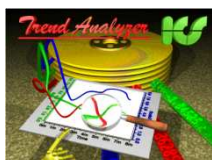
- Es una herramienta básicamente para extracción de datos básica.
- No existe suficiente documentación.

**Tabla II-13.** Valoración General Automated-SQL-Builder

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>		6	
	Puntaje ± 1	4	0	2			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	7	
	Puntaje ± 1	2.5	2.5	0	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>		5	
	Puntaje ± 1	4	1	0			

<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		0
	Puntaje ± 1	0	0	0		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		7
	Puntaje ± 1	5	0	2		

### 2.5.12. TrendAnalyzer



TrendAnalyzer permite a los usuarios finales mostrar y analizar tendencias. Esto proporciona una visualización fácil y rápida de las tendencias de los datos y tablas de documentación que fueron

exportadas de base de datos como Access, SQL-Server, Oracle, dBase o Excel.

Dispone de tres versiones Small Edition, Standard Edition y Professional Edition.

#### Características Principales

- Es una herramienta especializada para determinar tendencias y proyecciones
- Edición simultánea de varios diagramas.
- Las fuentes de datos pueden filtrarse directamente mediante instrucciones SQL.

#### Limitaciones

- No existe suficiente documentación.

**Tabla II-14.** Valoración General TrendAnalyzer

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>		6	
	Puntaje ± 1	4	0	2			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	6	
	Puntaje ± 1	2	2	2	0		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>		0	
	Puntaje ± 1	0	0	0			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		0	
	Puntaje ± 1	0	0	0			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		0	
	Puntaje ± 1	0	0	0			

### 2.5.13. Total Access Memo – ActiveX



Total Access Memo brinda la posibilidad de añadir el formato de texto enriquecido a Access. FMS' easy data binding, permite mostrar texto enriquecido en formularios e informes y almacenarlo en las tablas correspondientes.

#### Características Principales

- Esta diseñado específicamente para Microsoft Access que soporta el formato RTF.

**Tabla II-15.** Valoración General Total Access Memo

		Valoración				
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>		6
	Puntaje ± 1	4	0	2		
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	7
	Puntaje ± 1	2	2	1	2	
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>		6
	Puntaje ± 1	4	0	2		
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		7
	Puntaje ± 1	6	0	1		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		0
	Puntaje ± 1	0	0	0		

#### 2.5.14. Aspose AdHoc - .Net



Aspose.AdHoc es un componente para construcción de consultas para ASP.NET que se pueden utilizar para una gran variedad de tareas, que van desde los informes ad hoc para la búsqueda rápida y fácil en pantallas. Los desarrolladores pueden simplemente arrastrar y soltar el componente ad hoc a su formulario web, asignar un objeto DataTable como la fuente de datos y dejar que el componente haga el resto. El componente ad hoc utiliza un objeto DataTable como fuente de datos que permite utilizar prácticamente cualquier base de datos de fondo. El componente especial ofrece una variedad salidas que incluyen: sentencias SQL, sentencias: Select, Where, order by, Filtrado de datos y entre otros.

#### Características Principales

- Soporta aplicaciones .Net.
- Soporte para drag and drop.
- La fuente de datos puede ser un DataTable. Soporta múltiples tablas.
- Soporta alias para los campos de las tablas.
- Soporta hojas de estilo personalizadas.
- Aspose tiene varios productos que dan soporte para presentación de Reporting Services en formatos adicionales, así como el reconocimiento de códigos de barras.

#### Limitaciones

- Las consultas soportadas son sencillas, generalmente arrastre de campos de una tabla a un control de presentación formado por filas y columnas.
- No soporta agrupaciones.
- No es una herramienta para creación de reportes sofisticados.

**Tabla II-16.** Valoración General Aspose AdHoc - .Net

		Valoración				
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>		7
	Puntaje ± 1	4	2	1		
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	7
	Puntaje ± 1	2	2	1	2	
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>		6
	Puntaje ± 1	4	0	2		
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		6
	Puntaje ± 1	4	1	1		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		7
	Puntaje ± 1	5	0	2		

#### 2.5.15. ComponentOne Studio



Añade grids, presentación de informes, gráficos, planificación, manipulación de datos, interfaz de usuario y más para aplicaciones de dispositivos móviles y ActiveX de .Net y ASP.NET. Tiene las versiones Enterprise, .Net y Active X.

#### Características Principales

- Parte de la suite es el componente C1ReportDesigner, que crea reportes estilo Microsoft Access para Visual Studio .Net, así como también el componente C1PrintDocument y C1PrintPreview, que añade vista previa, formateo, impresión y exportación de las aplicaciones.
- El assembly C1Report

#### Limitaciones

- Cuando se usa importación, el modelo de objetos de C1Report no está disponible y los eventos de los reportes no son invocados. De allí la principal limitación de la importación, los reportes dependientes de manejadores de eventos de C#/VB no pueden ser renderizados al ser importados.
- Las propiedades Section, Font y Subreport son readonly.

- Propiedades como LineSpacing, SubreportHasData y LinkValue no se soportan, tampoco ColumnLayout o LabelSpacing y OverlayReplacements.
- El encabezado y el pie de página no soporta campos de la fuente de datos.

**Tabla II-17.** Valoración General ComponentOne Studio

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			6
	Puntaje ± 1	4	0	2			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		6
	Puntaje ± 1	2	2	0	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			6
	Puntaje ± 1	4	0	2			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			7
	Puntaje ± 1	4	1	2			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			0
	Puntaje ± 1	0	0	0			

#### 2.5.16. DXperience



Constituye toda una suite de componentes Visual Studio. NET, librerías, herramientas IDE producidos por Developer ExpSSRS en un solo paquete.

DXperience incluye Developer ExpSSRS. Entre sus librerías consta la de presentación de reportes XtraPrinting y XtraReports Suite, herramientas flexibles para la elaboración de informes de impresión.

#### Características Principales

- Contiene vista previa inmediata, todas las páginas de reportes son construidos en hebras separadas y son presentadas a la vista previa de impresión tan pronto como estén listas.
- Esta totalmente integrado con el IDE de Visual Studio .Net.
- XtraReports Suite soporta el formato condicional utilizado en Excel.
- El control PropertyGrid de la suite XtraVerticalGrid se utiliza para modificar propiedades del control en el diseñador del usuario final.
- Contempla más de 130 controles para medidores.

## Limitaciones

- Presenta un tanto de complejidad y su modelo de objetos no es del todo intuitivo, sin embargo son limitaciones que a medida que se familiariza con la herramienta resultan relevantes.
- Aunque la herramienta es bastante buena y supera a muchas de las mencionadas hasta ahora, comparado con Crystal Reports su presentación es aún algo insuficiente, aunque la habilidad para responder a eventos y la escritura de código como la presentación del reporte compensa otras limitaciones.

**Tabla II-18.** Valoración General DXperience

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			7
	Puntaje ± 1	4	1	2			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		7
	Puntaje ± 1	2.5	2.5	1	1		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			7
	Puntaje ± 1	4	2	1			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			8.5
	Puntaje ± 1	4.5	2	2			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			6
	Puntaje ± 1	5	0	1			

### 2.5.17. Data Dynamics

#### 2.5.17.1. ActiveReports for .Net



Genera una amplia variedad de informes de aplicaciones .NET (ahora incluye Visual Studio 2008 de Apoyo). ActiveReports. NET

está escrito en C# y proporciona integración con el IDE de Visual Studio. NET. Esto permite a los programadores de Visual Studio. NET aprovechar su lenguaje de programación (C # o Visual Basic. NET) en la programación de ActiveReports. Se basa en un esquema de licencias por desarrollador. ActiveReports .NET viene con un asistente para creación de informes y un asistente de conversión de reportes de Microsoft Access.

#### Características Principales

- Provee utilitarios para importación de Crystal y Microsoft Access.
- Soporta campos de texto enriquecido.

- Soporta construcción y modificación dinámica de reportes.
- La publicación se realiza mediante un motor que es administrado por assembly con strongname.
- El rendimiento en comparación con Crystal Reports es superior.

### Limitaciones

- Dado que la opción de enviar el reporte activo por mail utiliza un control ActiveX, no puede trabajar sobre navegadores como Firefox, adicionalmente este ActiveX utiliza directamente Microsoft Outlook. Por ello es recomendable guardar el reporte localmente, para luego enviarlo como archivo adjunto.
- No está integrado con el IDE de Visual Studio y no tiene un sistema de ayuda integrado.
- El viewer para aplicaciones web solo está disponible en la versión Active Reports Profesional.
- La escritura de formatos condicionales en tiempo de ejecución se torna compleja debido a que tiene que programarse un método en código C#/VB para ello.

**Tabla II-19.** Valoración General ActiveReports for .Net

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			8
	Puntaje ± 1	4	2	2			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		7
	Puntaje ± 1	2.5	2.5	1	1		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			7
	Puntaje ± 1	4	2	1			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			8
	Puntaje ± 1	4.5	2	1.5			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			6
	Puntaje ± 1	5	0	1			

### 2.5.17.2. ActiveReports

Desarrollado también por Data Dynamics, genera una gran variedad de reportes para una aplicación, ActiveReports V2.0 tiene una completa integración con el entorno de programación de Visual Basic, y permite crear reportes enriquecidos sin scripts encriptados



o work-arounds. Características robustas en ambiente de diseño que incluyen el report wizard, soporte para DAO, ADO, y OLE DB, soporte para eventos VB. Los reportes son compilados en la aplicación VB para incrementar el rendimiento en runtime.

### 2.5.17.3. DynamiCube



Crea aplicaciones cliente / servidor dinámicas e interactivas, para la explotación y análisis de datos. DynamiCube permite a los usuarios

finales realizar minería de datos altamente iterativa, con control total de tiempo de ejecución de los datos, las características de presentación de informes, capacidad analítica y están disponibles bajo cada escenario de acceso.

#### Características Principales

- Integra la capacidad de manipulación de datos de DynamiCube OLAP y creación de informes en la base de datos de las aplicaciones existentes.
- DynamiCube soporta filtrado dinámico optimizado de datos, pivoting, ranking y roll up.
- Los cálculos, sumatorias y presentación de datos relacionales o multidimensionales es bastante rápida.
- Proporciona la capacidad de definir diferentes márgenes en tiempo de ejecución.
- Permite la integración de la manipulación de datos DynamiCube's OLAP y la presentación de informes con las aplicaciones de base de datos existentes.

#### Limitaciones

- Una de las principales limitaciones es que la interfaz de usuario no es lo suficientemente intuitiva.
- Los informes generados son elementalmente básicos.

**Tabla II-20.** Valoración General DynamiCube

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			7
	Puntaje ± 1	4	2	1			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		0
	Puntaje ± 1	0	0	0	0		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			6
	Puntaje ± 1	4	2	0			

<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		7
	Puntaje ± 1	4	2	1		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		0
	Puntaje ± 1	0	0	0		

#### 2.5.17.4. ContourCube



ContourCube es un componente OLAP de alto rendimiento el desarrollo rápido de aplicaciones cliente servidor, aplicaciones web

y aplicaciones de business intelligence. ContourCube permite a los usuarios finales de OLAP realizar análisis estadísticos utilizando los datos actuales de cualquier base de datos relacional. ContourCube proporciona una interfaz muy dinámica para el análisis interactivo de datos, análisis sofisticados, estadística, la presentación de informes y funciones de gráficos.

#### Características Principales

- La tecnología OLAP utilizada en Countour Reportes, permite ver los datos en diferentes ángulos, sumar, desplegar y contraer datos, aplicar filtros, realizar un análisis gráfico instantáneamente.
- Contour Reporter tiene una sencilla interfaz estilo hoja de cálculo. Cuando se trabaja con un reporte, se puede acceder a docenas de formas de reportes con tiempos aceptables.
- Se puede imprimir y guardar los reportes en varios formatos, Countour Microbue, Excel, Word, HTML, así como enviarlos por email.
- Para optimizar las salidas, Countour Reporter soporta tres tecnologías OLAP, ROLAP (Relational OLAP – construcción de reportes desde bases de datos relacionales), MOLAP (Mobile OLAP – apertura rápida de microcubos pre construidos) Y HOLAP (Hybrid OLAP – trabajar con reportes pre construidos con la actualización online desde bases de datos relacionales).

- Countour Reporter se presenta en tres ediciones para diferentes categorías de usuarios: Administrador, Autor y Usuario. El Administrador configura el sistema, el autor crea y publica los reportes y los usuarios consumen los reportes.


### Limitaciones

- Cada cliente debe adquirir una licencia de Countour Cube para poder acceder a los reportes.

**Tabla II-21.** Valoración General ContourCube

		Valoración				
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>		7
	Puntaje ± 1	4	2	1		
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	7
	Puntaje ± 1	2.5	2.5	2	0	
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>		7
	Puntaje ± 1	4	2	1		
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		7
	Puntaje ± 1	4	2	1		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		0
	Puntaje ± 1	0	0	0		

### 2.5.18. Clover

**Clover**  Clover es una herramienta para consulta Web iSeries y Business Intelligence, que permite a los usuarios finales y personal técnico diseñar y ejecutar en tiempo real reportes y consultas web en sus navegadores. Proporciona alrededor de 20 plantillas para creación de reportes y gráficos que facilitan escribir en los informes totales, subtotales, saltos de nivel, drill downs, resúmenes o detalles, la producción en línea detalles gráficos (barras, pastel, la línea, medidores, etc.) usando las funciones de SmartCharts, y exportación a Excel.

### Características Principales

- Funciona con iSeries o i5, SQL proporciona asistentes para construir relaciones simples o complejas entre archivos de forma intuitiva.
- Clover se integra sin problemas con WebSmart, herramienta de desarrollo de aplicaciones Web.

- Las capacidades de desarrollo web de Clover, permite la creación de consultas y reportes en tiempo real, paneles ejecutivos con drill downs, y ad hoc. Clover incluye usuarios ilimitados.
- Se integra con WebSmart, Free \* Nexus Portal para mayor seguridad y acceso controlado a Informes y SmartCharts para gráficos y KPIs.

### Limitaciones

- La creación de reportes ad hoc no resulta lo suficientemente intuitiva para el usuario final.
- La dependencia existente con Nexus Portal para establecer la seguridad de los informes.

**Tabla II-22.** Valoración General Clover

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			7
	Puntaje ± 1	4	2	1			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		7
	Puntaje ± 1	2	2	1	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			6
	Puntaje ± 1	4	2	0			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			7
	Puntaje ± 1	4	2	1			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			6
	Puntaje ± 1	5	1	0			

### 2.5.19. Enterprise Reporting Applications de Actuate

Actuate provee una plataforma de software para aplicaciones de Reporting empresariales. Actuate Enterprise Reporting Applications, combina paneles ejecutivos, web reports, ad hoc query, análisis de business intelligence, y hojas de cálculo electrónicas administradas en servidor para entregar una experiencia unificada al usuario.

#### 2.5.19.1. Actuate BIRT



Es una flexible herramienta Java para construir y publicar reportes provenientes de fuentes de datos desde las típicas bases de datos

SQL, a fuentes de datos XML, hasta de objetos Java en memoria. Actuate BIRT está basado en el código abierto del proyecto Eclipse Business Intelligence Reporting and Tools (BIRT).

### **Características Principales**

- Flexibilidad en formatos de salida de informes. Entrega en páginas simples o múltiples de PDF o HTML. Reportes en múltiples vistas y formatos de impresión utilizando parámetros, secciones múltiples, niveles de agrupación, charts, etc. El contenido del informe puede ser también ser fácilmente exportado a formato CSV.
- Construcción de reportes desde cualquier data source. El acceso y generación de reportes desde base de datos SQL vía JDBC, objetos Java en memoria, archivos de texto, XML, o EJB.
- Diseño de reportes mediante asistentes o con código Java. Se puede crear y controlar los reportes programáticamente con scripting y APIs provistas si es requerido
- Preserva el formato desde la pantalla al papel, los reportes entregados son impresos tal como se visualizaron en el browser.

### **Limitaciones**

- Al ser Software OpenSource, uno de los principales problemas del manejo de código es la falta de indemnización, lo que puede aumentar los riesgos legales, la falta de mantenimiento y soporte, por lo que ninguna entidad es responsable de abordar y resolver los defectos del producto en forma oportuna. Sin embargo en compensación Actuate BIRT ha proporcionado el beneficio de una suscripción anual que asegura rendición de cuentas por problemas y errores, la indemnización legal y experiencia en soporte.
- Luego de la desinstalación de productos Actuate se presentan problemas con fuentes de datos ODBC y tipos de archivos.

**Tabla II-23.** Valoración General Actuate BIRT

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			7
	Puntaje ± 1	4	2	1			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		7
	Puntaje ± 1	2	2	1	2		
<b>Formatos de Exportación WYSIWYG</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			6
	Puntaje ± 1	4	2	0			
<b>Reportes Ad Hoc</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			9
	Puntaje ± 1	4	3	2			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			7
	Puntaje ± 1	5	1	1			

### 2.5.19.2. Actuate Analytics



Dirigido a organizaciones que quieran proveer funcionalidad analítica de presentación de informes a los usuarios que no se sienten conformes con la presentación de reportes ad hoc o sofisticadas herramientas de análisis multidimensional. Actuate Analytics está compuesto por varios componentes, entre ellos Cube Viewer, un programa de interfaz que permite a los usuarios analizar cubos, definir y guardar los informes de cubos multidimensionales para el análisis offline de los mismos.

Actuate Analytics es una herramienta OLAP personalizable, que se ve bastante al igual que otros visualizadores OLAP. Funciona sobre las bases de roles de usuario y soporte de directorios LDAP (y Active Directory), de la misma manera que lo hacen los portales. Otro servicio útil es la capacidad de correr sin conexión. Por otro lado, están algunas características que no tiene, por ejemplo no tiene previsión de series de tiempo.

Hay también una opción de integración para la distribución de reportes Actuate's iServer y la plataforma de administración, denominado Analytics Option for iServer. Los usuarios pueden aprovechar la integración con iServer para generar cubos, mientras iserver por si mismo puede configurarse para almacenar en cache y entregar cupos e informes cubo a consumidores de las aplicaciones de presentación de informes Actuate.

Actuate Analytics está basado a tres componentes:

- Actuate Cube Viewer: Una interfaz fácil de usar que capacita a los usuarios para analizar cubos, definir informes y almacenar cubos multidimensionales para análisis sin conexión.
- Actuate Cube Designer: Un asistente de diseño que capacita a los usuarios a definir y crear cubos OLAP ligeros paso a paso.
- Actuate Analytics Option for iServer: Una componente basado en servidor que genera bajo demanda o cubos programados, almacena y entrega cubos e informes de cubos a aplicaciones de reportes empresariales.

#### **Características Principales:**

- Acceso con Un Clic: Provee acceso a los datos de los cubos con un clic desde cualquier hipervínculo soportado, incluyendo usuarios de páginas principales, reportes basados en Web, Acrobat y hojas electrónicas.
- Cubos personalizados: Contenido seguro y específico por usuario a los reportes específicos a la información empresarial pertinente.
- Interfaz personalizables: La interfaz puede incluir o excluir funcionalidad, consecuentemente el front end se hace a la medida de las capacidades del cada usuario.
- Análisis Mobile: Permite a los usuarios analizar desde cualquier lugar sin conexión, análisis remoto.

#### **Limitaciones**

- Para ejecutar informes cuya fuente de datos es XML, se debe instalar el driver XML correspondiente. Además los asistentes para establecer fuentes de datos, no presentan como opción XML.
- En la versión 9. Actuate Information Object Designer utiliza la versión ODA 3.0, mientras que Report Designer Professional y Spreadsheet Designer utilizan versión 1.3, para utilizar ambos elementos es necesario recompilar el driver utilizando la versión 1.3.

- El control image, AclImageControl, no soporta la propiedad Border.
- Si no se especifica un orden en particular, los datos resultantes presentan un ordenamiento impredecible.
- Luego de la desinstalación de productos Actuate se presentan problemas con fuentes de datos ODBC y tipos de archivos.

#### **2.5.20. Hyperion System 9 BI+**



Recientemente Oracle realizó la adquisición de Hyperion y sus productos. Es una plataforma de business intelligence que proporciona el espectro de administración de reportes que combina información operacional y financiera. Se puede personalizar alto volumen, reportes pixel-perfect para publicar decenas de miles de consumidores en la Web con rendimiento superior. Un modulo de presentación de reportes financieros especializados con formatos predefinidos y construido con inteligencia financiera deja reunir y publicar rápidamente con calidad de producción libros de informes para administración, control y regulación de producción-informe sobre la calidad libros para control de la gestión y presentación de solicitudes de regulación. Y para consultas y presentación de informes ad hoc, la línea de usuario de negocio puede crear sus propios reportes interactivos para monitorear su desempeño y determinar tendencias.

#### **Características Principales**

- El producto es fuerte en el campo de la funcionalidad analítica que es provista a través de un motor OLAP junto con la unificación de la interfaz, con funcionalidad general de metadatos, escalabilidad de la aplicación, diseño de reportes, e integración de Microsoft Office. Hyperion ofrece análisis y capacidades para presentación de informes sobre una plataforma integrada.
- Funcionalidad analítica robusta.



## Limitaciones

- En el manejo de fechas, la función range tiene como limitación que puede trabajar únicamente en el período de tiempo jerárquico.
- Las celdas no se pueden ocultar condicionalmente de forma individual.
- Cuando se abre un archivo PDF creado por la funcionalidad de exportación a este formato, se puede dar un error intermitente que hace que la pantalla se quede en blanco. Este error hace que Internet Explorer falle al cargar el plug in de Adobe Acrobat.
- No se puede embeber funcionalidad de J2EE en un sistema Hyperion System con JSP.
- No se puede utilizar caracteres especiales en un nombre.
- Los archivos cuyos nombres exceden los 255 caracteres no son válidos.

**Tabla II-24.** Valoración General Hyperion System 9 BI+

		Valoración				
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>		7
	Puntaje ± 1	4	2	1		
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	5
	Puntaje ± 1	2	2	1	0	
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>		6
	Puntaje ± 1	4	2	0		
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>		0
	Puntaje ± 1	0	0	0		
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		8
	Puntaje ± 1	5	1	2		

### 2.5.21. Cognos 8 Business Intelligence, Reporting



La presentación de informes es un elemento clave dentro de Cognos 8 Business Intelligence. La presentación de informes permite acceder a una lista completa de auto-servicio de tipos de reportes, es adaptable a cualquier fuente de datos y opera desde una única capa de metadatos para una variedad de beneficios tales como la presentación de informes multilingües.

#### Características Principales

- Un entorno de autoría para crear todos los tipos de informes, incluidos dashboards.

- Una consulta ejecutada sobre fuentes de datos de múltiples proveedores, incluso con un único objeto de reporting.
- Autoría automáticamente adaptativa a los ajustes de diseño desde que los objetos son añadidos, movidos o removidos.
- Aplicaciones embebidas, sitios Web y contenido no-BI dentro de un reporte.
- Drag-and-drop, incorpora datos, texto, cuadros, gráficos e imágenes.
- Edición de reportes con prompts y comandos toolbar.
- Visualizaciones avanzadas y capacidades de graficación a través de Cognos Visualizer
- Uso de una variedad de gráficos, matrices, barras bar/3D, pastel / rosquilla, línea, embudo, de dispersión, la densidad de puntos, cascada, etc.
- Crea diseños complejos y multi página usando diferentes fuentes de datos sin necesidad de programación o workarounds (soluciones alternas).
- Conectividad completa sin tomar en cuenta el entorno, bases relacionales, SAP® BW, Excel, XML, JDBC, LDAP y servicios Web.
- Soporte para Windows, UNIX, y sistemas operativos Linux, incluyendo plataformas mixtas.
- Exportación a múltiples formatos: Excel, PDF, XML, HTML, y CSV.


### **Limitaciones**

- Varias limitaciones en escalabilidad y desarrollo de reportes.
- La configuración de la seguridad es compleja.
- No soporta drill-through ni mapas de reportes en formato Excel, y
- No se puede separar consultas de tabla cruzada directamente, por ello deben utilizarse por separado.

**Tabla II-25.** Valoración General Cognos 8 Business Intelligence

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			8.5
	Puntaje ± 1	4.5	2	2			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		8
	Puntaje ± 1	2	2	2	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			8
	Puntaje ± 1	4.5	2.5	1			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			8
	Puntaje ± 1	4.5	2.5	1			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			7
	Puntaje ± 1	5	1	1			

### 2.5.22. Oracle reports

 Oracle Reports es una herramienta Oracle's enterprise-reporting. Proporciona un entorno estable para crear y distribuir reportes a sus usuarios a través de una casi interminable lista de canales de distribución.

Oracle reports viene con diseño de modelos flexibles que permiten crear informes optimizados tanto para salida paginada como impresión, así como la Web orientado a la producción. Basado en estándares, el entorno de presentación de informes puede ser extendido para solucionar las necesidades exactas de los clientes. Oracle Reports consta de dos componentes – Reports Builder y Reports Services. Reports Builder puede utilizarse para conectarse a cualquier fuente de datos relacional como base de datos, archivos de texto, XML, u OLAP.

Reports Services permite generar la salida del reporte en cualquier formato una vez que su informe ha sido diseñado. Por ejemplo, puede generar informes en PDF, RTF, XML, HTML, Excel y otros formatos. Estos informes pueden ser enviados a cualquier destino, como sistema de ficheros, correo electrónico, impresora o servidor FTP.

#### Características Principales

- Publicación de reportes en la Web: Datos formateados para ser publicados en la Web en formatos PDF; HTML (3.2), HTML Cascading Style Sheets (HTML 4.01), y XML. También se puede inyectar contenido dinámico dentro de la página Web existente.

- Asistente de reportes, para crear reportes rápidamente con los formatos usados habitualmente.
- Modo de diseño para crear nuevos estilos de reportes. También pueden ser creadas nuevas plantillas de reportes.
- Crear y embeber gráficos en los reportes.
- Incluir reportes en aplicaciones Java.
- Describir una definición completa de un reporte en XML.
- Características completas de seguridad y administración, junto con escalabilidad e integración con Microsoft Office.

### Limitaciones

- No soporta personalización para XML de archivos REP. La extensión REP, define a un archivo que contiene una única definición de reporte en formato binario. Son utilizados únicamente para ejecutar reportes no se puede editar un archivo .REP.
- La salida DelimitedData no permite que consultas múltiples sean enlazadas en una consulta principal, debido a que no se soportan grupos paralelamente.
- Cuando se exporta datos de una hoja de cálculo a Oracle Reports, no se soporta descubrimiento de gráficos ni porcentajes. Cuando se da un error en la exportación las alertas son el único medio para que el usuario determine la acción a tomar.
- Carencia de desarrollo y diseño robusto. No existe funcionalidad con respecto a la facilidad de uso.

**Tabla II-26.** Valoración General Oracle reports

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			7
	Puntaje ± 1	4	2	1			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		8
	Puntaje ± 1	2	2	2	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			8
	Puntaje ± 1	4.5	2.5	1			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			7
	Puntaje ± 1	4	2	1			

Reportes Ad Hoc	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>		7
	Puntaje ± 1	4.5	0	2.5		

### 2.5.23. MicroStrategy Enterprise Reporting



MicroStrategy's reporting technology, permite a las compañías crear un completo rango de formatos de reportes desde reportes de negocio clásicos hasta reportes operativos muy detallados, que están típicamente basados en papel. Uno de los aspectos más importantes y únicos de esta solución es la capacidad de combinar técnicas tradicionales de diseño de reportes en bandas usados en reportes operativos con diseño basado en modelos potentes gráficos usados para producir y scorecards y dashboards.

#### Características Principales

- Método drag and drop para creación de reportes.
- Excelente formato para calidad de presentación.
- Presentación de informes paramétricos. Los usuarios pueden obtener cualquier número de consultas previo a correr el reporte, permitiendo a sus consultantes determinar el contenido que será desplegado en el reporte. El prompting avanzado basado en un motor de consulta que permite a los usuarios elegir y seleccionar el contenido de un reporte completamente independiente del diseño del mismo.
- Contenido personalizado automáticamente: Los administradores de BI solo necesitan crear un reporte que el sistema de forma automática lo presentará en diferentes vistas adecuadas para cada usuario. Múltiples adaptaciones de este reporte son entonces generadas y distribuidas basados en el rol de cada usuario y grupo de afiliación.
- Interfaz de usuario personalizada acorde el perfil de usuario.
- Con una sencilla interfaz basada en Web el usuario obtiene una presentación de informes empresarial, análisis de cubos, análisis adhoc, facilidades de minería de datos.

- Fuertes capacidades de OLAP, con acceso extendido a cubos SAP BW, junto con una robusta administración y capacidades de seguridad.

**Limitaciones**

- La administración no se basa en la web.
- Existe baja escalabilidad en las novedades del producto.
- Incluye el componente Report Services dentro del core del producto, sin embargo, las limitación en acceso a datos aminoran el encanto de la producción mediante este mecanismo.

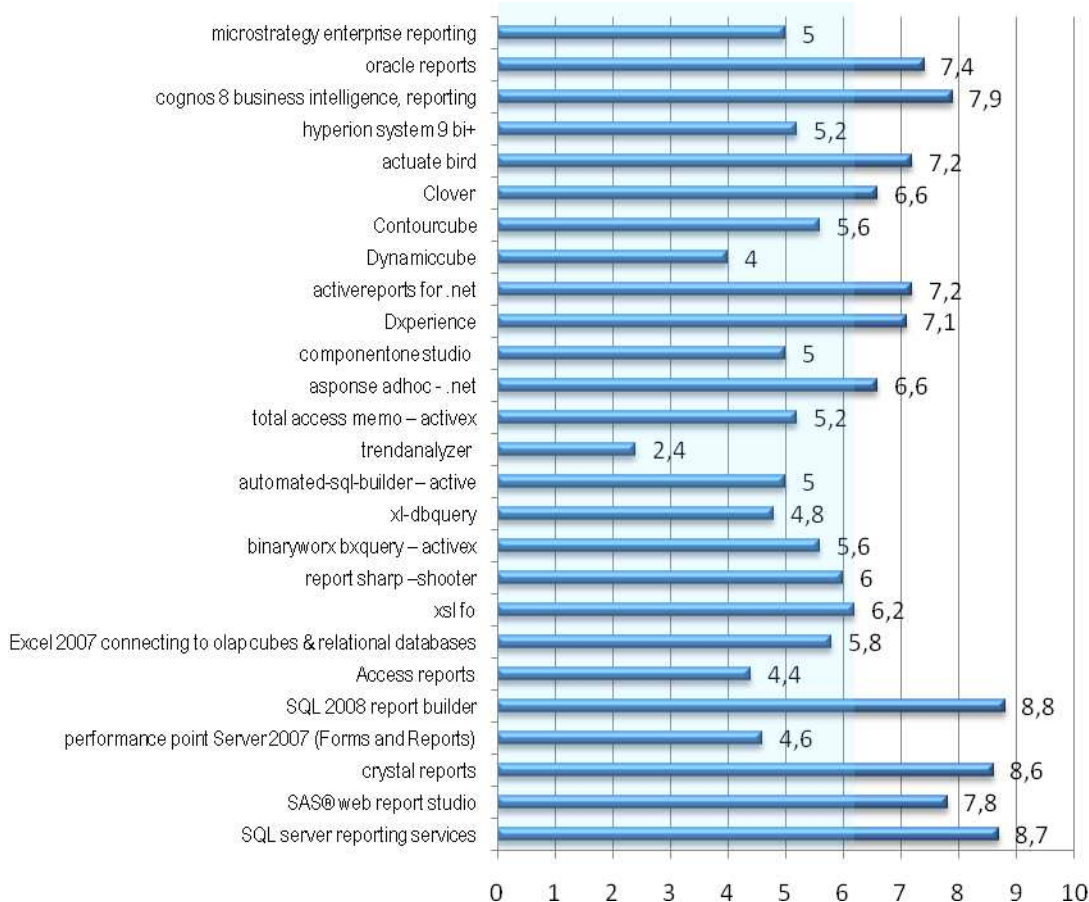
**Tabla II-27.** Valoración General MicroStrategy Enterprise Reporting

		Valoración					
<b>Facilidad de Uso</b>	Parámetro	<b>FU1</b>	<b>FU2</b>	<b>FU3</b>			6
	Puntaje ± 1	4	2	0			
<b>Personalización</b>	Parámetro	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>		6
	Puntaje ± 1	2	2	0	2		
<b>Formatos de Exportación</b>	Parámetro	<b>FE1</b>	<b>FE2</b>	<b>FE3</b>			6
	Puntaje ± 1	4	2	0			
<b>WYSIWYG</b>	Parámetro	<b>W1</b>	<b>W2</b>	<b>W3</b>			0
	Puntaje ± 1	0	0	0			
<b>Reportes Ad Hoc</b>	Parámetro	<b>AH1</b>	<b>AH2</b>	<b>AH3</b>			7
	Puntaje ± 1	4.5	0	2.5			

2.6. Cuadro de valorización de PEI.

Tabla II-28. Cuadro de valorización de herramientas.

<i>TECNOLOGIA</i>	<i>Promedio</i>	<i>Facilidad de Uso</i>	<i>Personalización</i>	<i>Formatos de Exportación</i>	<i>WYSIWYG</i>	<i>Reportes Adhoc</i>
<i>SQL server reporting services</i>	↑ 8.7	9	8.5	8	9	9
<i>SAS® web report studio</i>	↑ 7.8	9	8	7	7	8
<i>crystal reports</i>	↑ 8.6	9	8	9	9	8
<i>performance point Server 2007 (Forms and Reports)</i>	→ 4.6	6	5	5	7	0
<i>SQL 2008 report builder</i>	↑ 8.8	9	9	8	9	9
<i>Access reports</i>	↓ 4.4	6	5	5	6	0
<i>Excel 2007 connecting to olap cubes &amp; relational databases</i>	→ 5.8	6	7	0	8	8
<i>xsl fo</i>	→ 6.2	7	7	9	8	0
<i>report sharp –shooter</i>	→ 6	7	8	8	7	0
<i>binaryworx bxquery – activex</i>	→ 5.6	7	7	0	7	7
<i>xl-dbquery</i>	→ 4.8	6	6	5	0	7
<i>automated-sql-builder – active</i>	→ 5	6	7	5	0	7
<i>trendanalyzer</i>	↓ 2.4	6	6	0	0	0
<i>total access memo – activex</i>	→ 5.2	6	7	6	7	0
<i>asponse adhoc - .net</i>	→ 6.6	7	7	6	6	7
<i>componentone studio</i>	→ 5	6	6	6	7	0
<i>Dxperience</i>	↑ 7.1	7	7	7	8.5	6
<i>activerreports for .net</i>	↑ 7.2	8	7	7	8	6
<i>Dynamiccube</i>	↓ 4	7	0	6	7	0
<i>Contourcube</i>	→ 5.6	7	7	7	7	0
<i>clover</i>	→ 6.6	7	7	6	7	6
<i>actuate bird</i>	↑ 7.2	7	7	6	9	7
<i>hyperion system 9 bi+</i>	→ 5.2	7	5	6	0	8
<i>cognos 8 business intelligence, reporting</i>	↑ 7.9	8.5	8	8	8	7
<i>oracle reports</i>	↑ 7.4	7	8	8	7	7
<i>microstrategy enterprise reporting</i>	→ 5	6	6	6	0	7

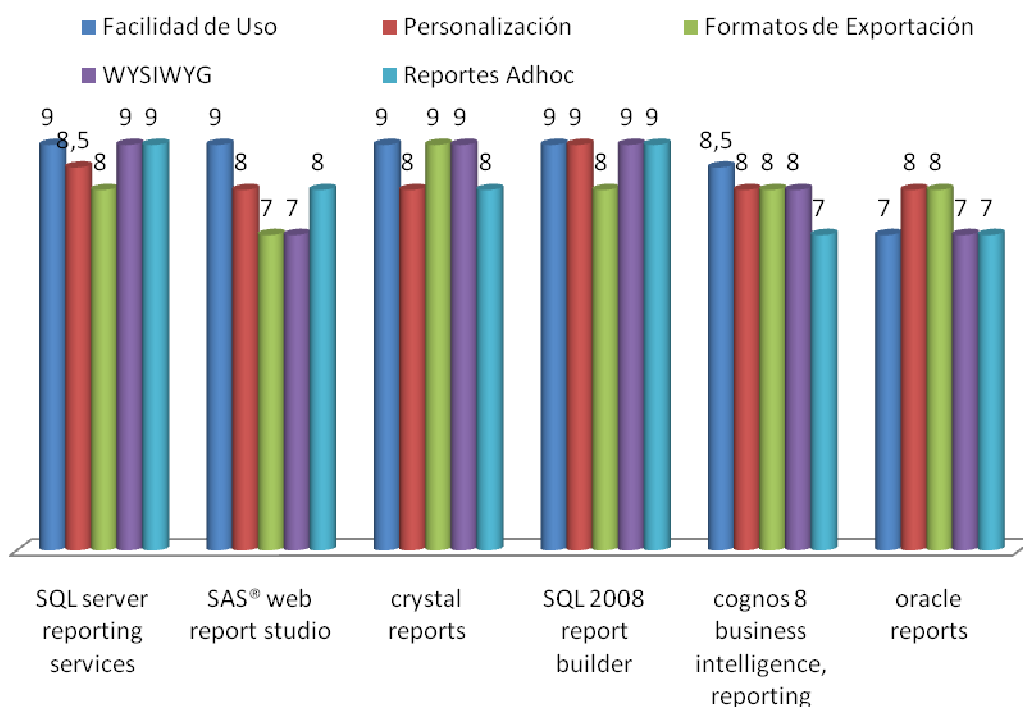


**Figura II. Promedio de valoración PEIs**

En el cuadro de valorización de factores relevantes de una herramienta para la elaboración de reportes, indudablemente Crystal Reports y SQL Reporting Services contienen las mejores marcaciones para las características analizadas. Es por ello y debido también a que estas dos tecnologías son precisamente las que lideran el mercado, que a partir de este punto el presente estudio se enfocará enteramente a estas dos plataformas de diseño y elaboración de informes.

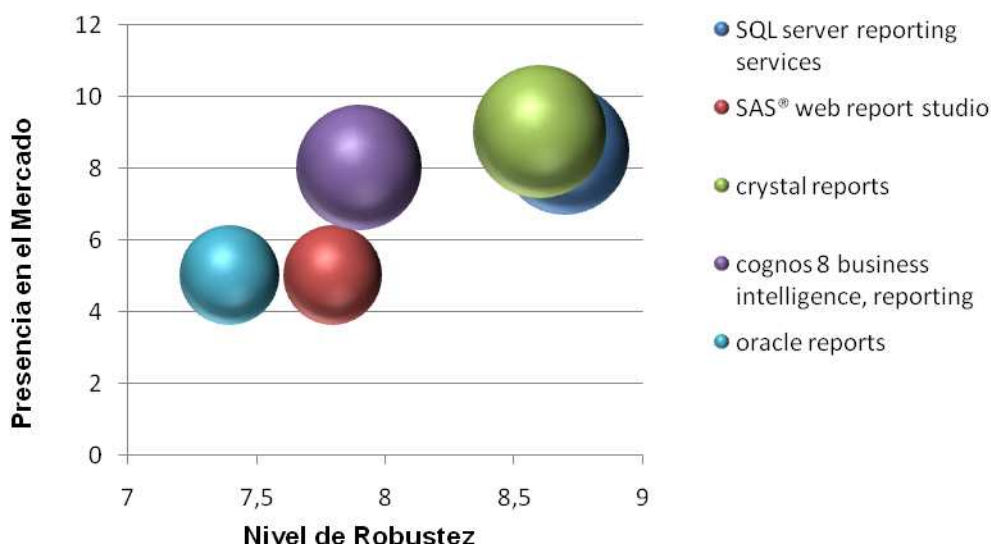
La siguiente es una gráfica comparativa acorde a la valoración de las diversas herramientas evaluadas.





**Figura III.** Valoración de las herramientas más robustas para elaboración de informes

Se ha realizado un sondeo de la presencia en el mercado de las plataformas cuyo promedio de valoración es el más alto, obteniendo los resultados se la siguiente figura.



**Figura IV.** Presencia en el mercado de plataformas de reporting más robustas.

## **2.7. Crystal Reports**

Crystal Enterprise fabricado originalmente por BusinessObjects, recientemente adquirido por SAP, es una arquitectura de acceso a datos, informes y entrega de informaciones que reduce el tiempo de respuesta a cualquier solicitud de informes o desarrollo de aplicaciones.

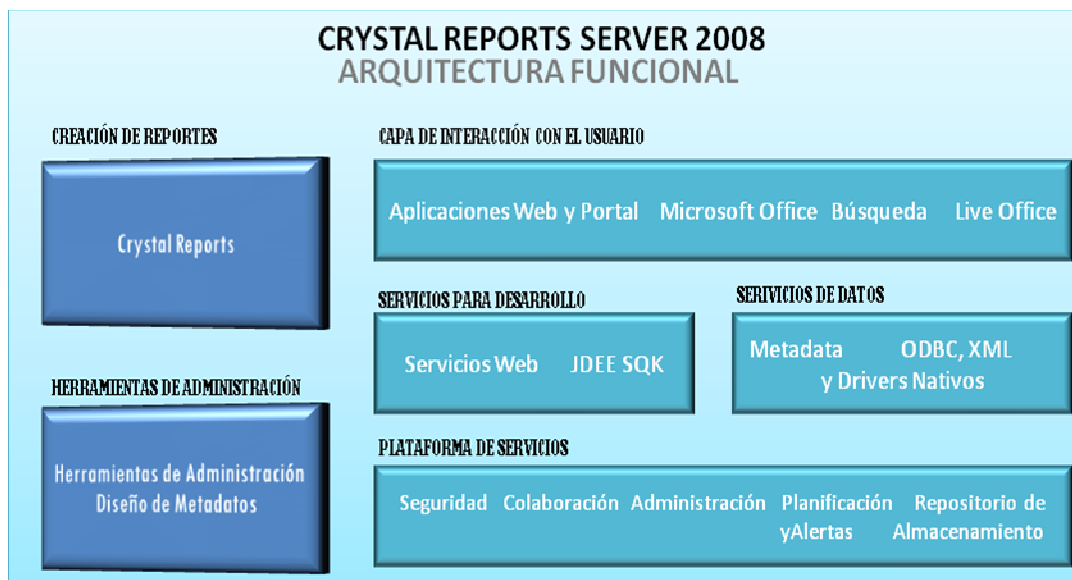
El corazón de la plataforma de Crystal Enterprise es una robusta infraestructura multiservidora y multiplataforma para organizar, gestionar y suministrar información a través de una organización. Crystal Enterprise se integra a la perfección con una diversidad de datos de nivel industria, red, aplicación y sistemas de seguridad.

Crystal Enterprise suministra un depósito central para organizar contenido, asegurar que los usuarios puedan acceder fácilmente a la última información, así como historiales instantáneos. Un sistema de seguridad jerárquico le permite imponer políticas consistentes acerca de la visualización de información e interacción. Servicios especializados manejan la generación de informes o acceso de datos bajo demanda o bien planificación de informes basada en acontecimientos y tiempo. Capacidades de distribución construidas interiormente pueden alertar a los usuarios automáticamente vía correo electrónico y portales, o informes de proyección externa en diversidad de formatos para destinos específicos (p. ej. correo electrónico, sitios FTP, impresoras en línea).

Crystal Enterprise se convierte en parte de la infraestructura interna de IT, mediante el suministro de las más altas cuotas de acceso concurrente y la fiabilidad de los cinco nueves (99'999% - confirmado en la prueba de terceras partes) que se espera de una solución de empresa.

### **2.7.1. Arquitectura**

Crystal Reports es una tecnología para creación e integración de reportes con datos provenientes de múltiples fuentes.



**Figura V.** Arquitectura Funcional de Crystal Reports 2008

Crystal Enterprise fue diseñado para integrar de forma simple los datos preexistentes, Web y los aplicativos de negocios, sin imponer nuevos patrones o procesos. Crystal Enterprise es una plataforma para la elaboración de informes, análisis y suministro de información, que ayuda a las compañías a hacer un mejor uso de los datos de la empresa para entender y mejorar su desarrollo comercial.

Con Crystal Enterprise, se puede:

- Transformar casi cualquier dato en importante contenido interactivo.
- Analizar información para adquirir comprensión sobre tendencias y puntos de interés significativos en el desempeño comercial.
- Gestionar el suministro de información orientada e interactiva a miles de usuarios a través de la Web, correo electrónico o aplicaciones personalizadas

### **2.7.2. Interfaces de desarrollo**

Un juego completo de interfaces de usuario de bandeja de salida con un código de fuentes accesible da la capacidad de adaptar de manera personalizada a una organización la apariencia y esencia de Crystal Enterprise. También se puede acceder a la funcionalidad

de Crystal Enterprise a través de profusos interfaces de desarrollo para entornos Java, NET y COM. Se da la posibilidad de integrar contenido dentro de un portal corporativo, automatizar tareas repetitivas de tipo administrativas, o construir potentes aplicaciones analíticas basadas en la red para afrontar las necesidades específicas del usuario.

### **2.7.3. Creación de contenidos y acceso a los datos**

Crystal Enterprise integra a Crystal Reports y Crystal Analysis, ambas herramientas de creación de informes ganadoras de premios, para permitir la creación rápida de contenidos interactivos virtualmente desde cualquier fuente de datos. Los diseñadores de informes pueden compartir y reutilizar voluntariamente piezas comunes de acceso de datos, edición y lógica de negocios para acelerar los ciclos de desarrollo. Lenguajes ricos en fórmulas y controles de planos en cada herramienta permiten un control extensivo sobre la manipulación de datos y el diseño de los informes.

Diseño integral y opciones de formato Crystal Reports provee más opciones de conectividad a datos que cualquier otra herramienta. Incluye más de 30 drivers para acceso a bases de datos relacionales, fuentes de datos XML y cubos OLAP (Incluyendo sistemas ERP, CRM, Oracle, IBM DB2 y Microsoft SQL Server). También puede acceder a datos personalizados a través de JavaBeans y objetos COM (ADO record sets) para una conectividad más flexible.

### **2.7.4. Informes**

Crystal Enterprise suministra toda una exhaustiva serie de características para la visualización, la navegación y la interacción con informes editados. Crystal Enterprise permite a los usuarios penetrar efectivamente en los informes, interactuar con gráficos y buscar informes grandes para descubrir puntos de interés significantes. Los informes también se pueden exportar a toda una variedad de formatos (incluyendo Excel, PDF y Word) para su uso autónomo o su integración en otros documentos.

### **2.7.5. Análisis**

La funcionalidad analítica sofisticada se construye directamente dentro de la plataforma de Crystal Enterprise. Proporciona una fácil creación y modificación de informes con una guía de trabajo que permite a los trabajadores, clientes y proveedores obtener y compartir datos de negocio y les permite tomar decisiones inteligentes. Crystal Enterprise facilita un espectro completo de capacidades analíticas, desde la modificación simple de informes al diseño o edición detallada de informes para análisis en profundo permitiendo a los usuarios en potencia y usuarios finales compartir efectivamente sus datos.

### **2.7.6. Repositorio Crystal**

El nuevo Repositorio Crystal permite almacenar elementos clave de los reportes de una solución, tales como objetos de texto, imágenes, sentencias SQL y funciones personalizadas. Gracias a este repositorio central se puede reutilizar estos objetos en múltiples reportes. Este almacén centralizado de objetos permite minimizar los esfuerzos de mantenimiento de sus reportes y al mismo tiempo ser más productivo en el diseño de reportes nuevos.

De la misma manera, se puede crear plantillas corporativas para ser aplicadas a todos los reportes de una organización y así evitar dar formato una y otra vez a cada uno de los nuevos reportes que se crea.

La interacción y acceso a los datos que sus usuarios demandan Crystal Reports habilita la visualización e interacción con los reportes a través de una amplia variedad de dispositivos y entornos. Los Report Parts (fracciones de reporte) o el soporte para Smart Tags en Office XP, se utilizan para proveer acceso, navegación e interacción con los reportes vía portales web,

Los visores Web avanzados habilitan a los usuarios finales para realizar búsquedas dentro de los datos de un reporte y exportarlas posteriormente a Microsoft Excel, Word y páginas HTML con el vínculo dinámico al reporte original.

### **2.7.7. Características Principales**

- **Control SQL** – Se puede editar las sentencias SQL directamente para un control ilimitado sobre la conectividad con la base de datos.
- **Repositorio** – Se puede almacenar objetos de reporte clave en una librería centralizada, para ser reutilizados en otros reportes.
- **Report Application Server** - Servicio de procesamiento de reportes basado en Web, para realizar integración de contenido dinámico en aplicaciones Web empresariales.
- **SDKs para .NET, Java y COM** – Se puede crear aplicaciones utilizando cualquier plataforma de desarrollo y controlando la visualización, interacción y modificación de los reportes en tiempo de ejecución.
- **Licencias de publicación Web flexibles** - Capacidades de procesamiento en cola, que hacen a un mejor aprovechamiento de licencias

Crystal Reports está disponible en 3 ediciones para satisfacer las diferentes necesidades de los desarrolladores, profesionales de TI y usuarios finales.

### **2.7.8. Servicios sobre la plataforma Crystal Decisions**

Para asegurar que cada usuario aproveche al máximo las herramientas de Crystal Decisions y maximice el retorno de la inversión, Crystal Solutions ofrece un amplio rango de soluciones de entrenamiento personalizado, consultoría y servicios de soporte técnico, prestados por especialistas de producto quienes cuentan con amplia experiencia y certificaciones internacionales.

Crystal Decisions es una empresa líder en el mercado en la elaboración de reportes, análisis y distribución de información.

## **2.8. Microsoft Reporting Services**

SQL Server 2008 Reporting Services es una solución basada en el modelo cliente servidor, donde se puede crear, administrar y distribuir informes basados en Web de forma interactiva. Para muchas compañías el reto de obtener la información correcta, en el tipo y

a la gente correcta, con un amplio rango de habilidades y experiencia, a quien necesita acceder a datos empresariales. SQL Server 2008 Reporting Services extiende la plataforma Microsoft Business Intelligence (BI), a todo el personal que requiere acceder a los datos. Reporting Services es un ambiente empresarial administrador vía Web Services. Los reportes pueden ser entregados en una variedad de formatos con un amplio rango de interactividad y opciones de impresión.

Pueden ser realizados análisis complejos por una gran extensión de audiencia a través de la distribución de reportes como fuente de datos se los procesos propios de business intelligence de una empresa. Siendo un componente integrado de SQL Server 2008, Reporting Services provee:

- Un alto rendimiento en la ingeniería para el procesamiento y formato de reportes.
- Un completo conjunto de herramientas para creación, administración, y visualización de reportes.
- Una arquitectura extensible e interfaces abiertas para reportes embebidos o integración con soluciones de reportes de diversos entornos IT. Los escenarios claves de Reporting Services combinan una sencilla y completa plataforma de reportes con una arquitectura escalable y flexible para satisfacer una amplia variedad de necesidades en informes.
- Las empresas pueden utilizar Reporting Services para sus reportes operativos o aplicación de business intelligence. Usando reporting services el staff de una corporación puede diseñar una variedad de reportes y direccionarlos a la totalidad del personal de la empresa, usando una combinación de distribución vía e-mail, y publicación sobre un portal corporativo.

### **2.8.1. Arquitectura**

La nueva arquitectura de reporting services fue cambiada fundamentalmente por la remoción de las dependencias hacia Internet Information Services (IIS) y la consolidación de las aplicaciones en un único servicio. A partir de ahora ya no será necesario usar IIS

para poder disponer de toda la funcionalidad de SSRS y las tareas de mantenimiento se ven reducidas significativamente.

La nueva arquitectura embebe las capacidades del servidor HTTP en el mismo report server, permitiendo correr un report server como una verdadera aplicación en tres capas, separado del Front-End de las aplicaciones web que lo utilizan a acceder a él. El servidor de reportes también incluye una nueva capa de hosting de autenticación, administración memoria y procesos internos y soporte de tracing y logging end-to-end. Tiene soporte nativo para HTTP.SYS y ASP.NET.

Reporting Services alberga a las tecnologías ASP.Net y al Microsoft .Net Framework que son construidas en el CLR de SQL Server y aprovecha las capacidades HTTP.SYS del sistema operativo.

El servidor de reportes incluye un listener HTTP que acepta las peticiones que son dirigidas a un URL y pero que se define durante la configuración del servidor.

Aunque a nivel de arquitectura se presente como un único servicio, los diferentes componentes se encuentran accesibles a través de URL, por HTTP, tal como en la versión 2005.

Si SSRS se encuentra integrado con SharePoint el Report Manager no es accesible.

Como parte de los procesamientos en background, los servicios de programaciones, suscripciones y entrega – envió de reportes tienen un módulo independiente de administración de memoria y del método de autenticación.

Funcionalmente se mantiene la misma interacción con la que SSRS se lanzó al mercado, la siguiente figura describe dicha funcionalidad.



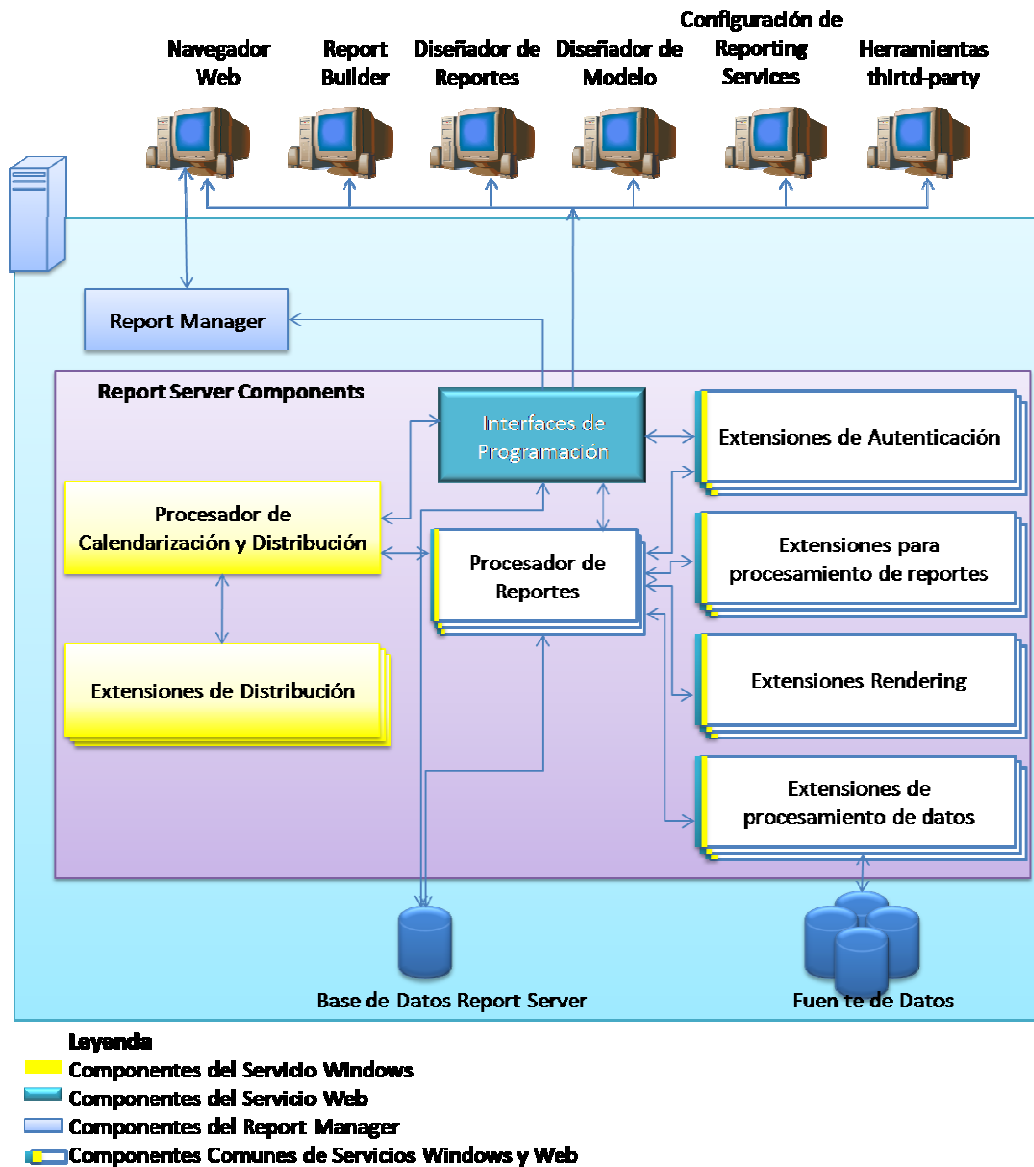


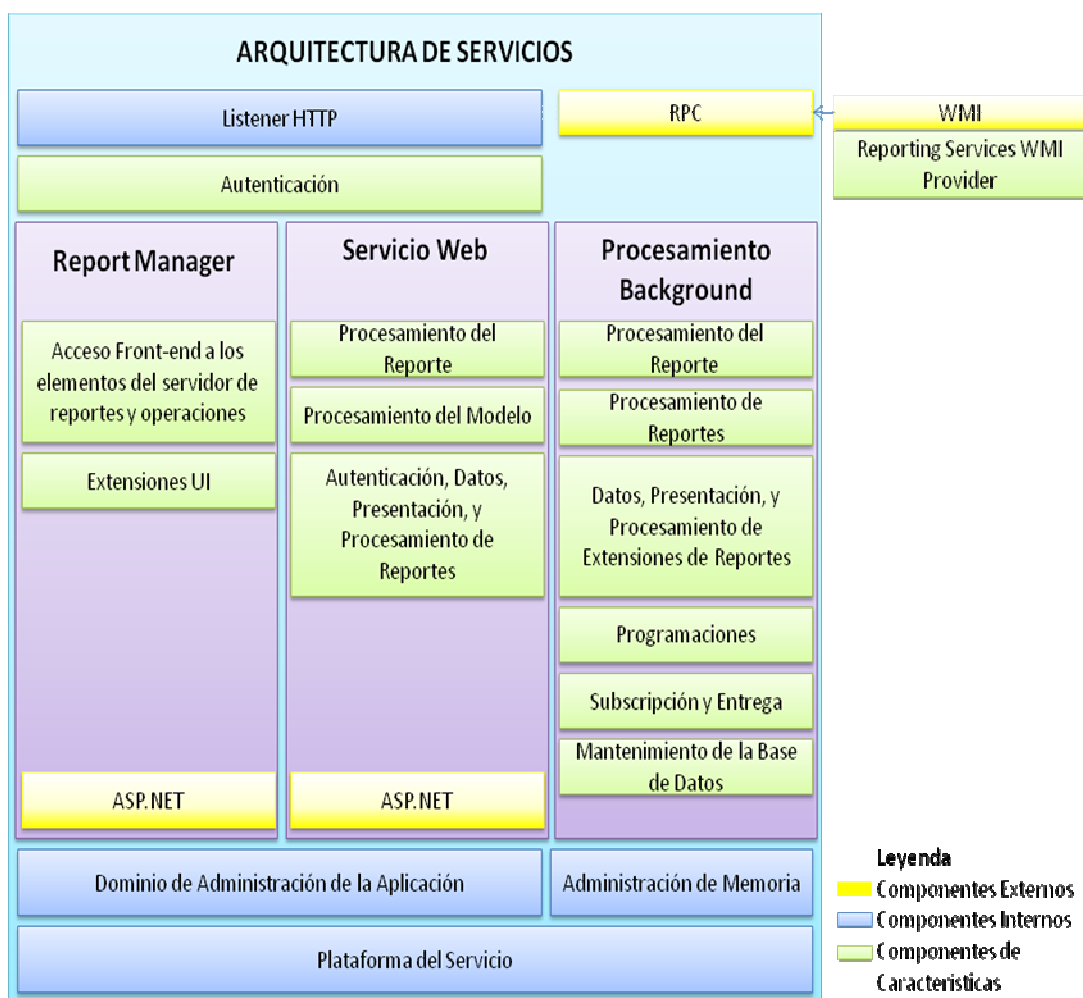
Figura VI. Arquitectura Funcional de SSRS.

### 2.8.2. Servicios e Infraestructura

El servicio Windows Report Server es un conjunto de aplicaciones consolidadas que corren en un proceso único, bajo una cuenta única con acceso a una única base de datos y grupo de archivos de configuración. Las opciones de configuración para el servicio en su totalidad están almacenadas en los archivos RReportServer.config, ReportServerServices.exe.config, y la base del servidor de reportes.

Con el servicio, tanto Report Manager como el Servicio Web y el procesamiento en background corren en dominios de aplicación separados. Aunque las tres áreas estén habilitadas por omisión, se puede configurar que partes del servicio estén disponibles en cualquier momento dado. Por ejemplo, se puede desactivar el servicio web si no se desea soportar reportes bajo demanda o interactivos.

Todas las áreas del servidor están aisladas en dominios de aplicación separados, la administración de memoria y la salud de los procesos son administradas como un todo.



**Figura VII.** Arquitectura de Servicios SSRS.

En el diagrama no constan todos los componentes de la arquitectura, por ejemplo, la inicialización y encriptación reversible así como operaciones críticas del servidor que existen por fuera del ámbito de las tres áreas ilustradas en la figura.

### **2.8.3. Autenticación y procesamiento de memoria para procesamiento en Background**

Como se puede ver en la Figura VII. Arquitectura de Servicios SSRS., el procesamiento background maneja la autenticación y administración de memoria de modo diferente del que se utiliza para Report Manager y el Servicio Web. Los procesos en background usan Authz.dll para verificar si la cuenta del usuario que fue usada para crear la suscripción, todavía tiene los permisos para ver el reporte. Esta verificación asegura que el usuario quien va a recibir el reporte, sea un usuario Windows válido en el dominio. Todos los otros procesamientos de reportes y modelos que se ejecutan como un proceso background requieren la identidad de una cuenta de ejecución desatendida.

### **2.8.4. Reporting Services WMI Provider**

El proveedor Reporting Services WMI está construido sobre la tecnología estándar Windows Management Instrumentation (WMI) para control de acceso y mantenimiento de información en un ambiente empresarial. WMI para SSRS sirve como un proveedor de instancia, mapea los elementos de la configuración XML del servidor de reportes a un conjunto de clases, las cuales incluyen métodos con información de configuración del report server que pueden ser añadidos, removidos, o modificados.

Las clases WMI están incluidas en Reporting Services para permitir que los componentes report server y report manager sean controlados tanto local como remotamente, para proveer una forma de descubrir cuales máquinas en la red están ejecutando un servicio web de Report Server, y para activar una instancia en una implantación scale-out.

Los administradores del sistema y administradores de la base de datos pueden utilizar dichas clases para hacer cambios a configuraciones de el report server y Report Manager después de la instalación, o ejecutar tareas administrativas locales o remotas, estas tareas pueden incluir modificación de credenciales de conexión a base de datos entre el report server y la base de datos report server, cambio del nombre de la base de datos, y cambios del URL que define la ruta para las instancias de report server o Report Manager. MSReportServer.Instance.Class y MSReportServer.ConfigurationSetting.Class son dos de las clases disponibles.

### **2.8.5. Novedades en Programación de Report Server**

SQL Server 2008 Reporting Services introduce nuevos métodos y capacidades de pre procesamiento para definiciones de reportes. Como la extensión Report Definition Customization (RDC), que se puede utilizar para personalizar la definición de un reporte, especificando identidad, cultura, basado en los valores que se detectan en el RDL, antes que sea pasado al motor de procesamiento. Esto es especialmente útil en aplicaciones que soportan múltiples culturas. Con RDC se puede crear las interfaces IReportDefinitionCustomizationExtension, IReportContext, IUserContext, IParameter, AuthenticationType, RdceCustomizableElementId.

Esta versión también incluye nuevos métodos para la clase ReportService2006, que provee acceso programático para ejecutar a report server que se ejecuta en modo integrado con SharePoint. Entre los métodos que figuran constan ListJobs, CancelJobs, CreateDataDrivenSubscription, CreateDataDrivenSubscriptionProperties.

### **2.8.6. Model Designer en Business Intelligence Development Studio**

En un proyecto Report Model, se puede definir fuentes de datos, diseñar vistas para fuentes de datos y crear un modelo desde la vista de la fuente de datos, es posible reorganizar los elementos del modelo como entidades o perspectivas.

Los modelos de reportes son utilizados en Report Builder para la creación de reportes Ad Hoc.

### **2.8.7. Administración de informes Reporting Services en Modo Sharepoint**

SSRS 2008, tiene dos modos de instalación, por ficheros y en modo integrado con SharePoint. Con este último, informes, orígenes de datos y modelos de informes se publican en librerías de Sharepoint. Al abrir un informe en WSS, el Web Partvisor de informes llama al API de ReportServer para procesar y renderizar el informe. Los usuarios pueden administrar las propiedades de los informes y suscribirse a ellos a través del interfaz de WSS (llama al API SOAP RS). La interfaz de usuario permite lanzar Report Builder para crear y editar informes.

### **2.8.8. AdHoc reporting**

SQL Server 2008 Reporting Services incluye una nueva herramienta para reportes ad hoc que habilita a los usuarios de la empresa a crear sus propios reportes y explorar los datos corporativos. Report Builder incorpora un modelo de consultas amigable que habilita a los usuarios para que puedan construir sus reportes sin entender a profundidad conceptos técnicos de acceso a datos.

### **2.8.9. Embedded reporting**

Los vendedores de software independiente (Independent Software Vendors ISV), pueden usar Reporting Services para proveer reportes predefinidos o reportes ad hoc, como parte de un paquete de una aplicación. Los clientes de IT de una organización pueden acceder a esos reportes as-is, reportes personalizados o crear nuevos para necesidades específicas de una empresa.

### **2.8.10. Web-based reporting for partners/customers**

Las organizaciones pueden desplegar reportes interactivos basados en web a fin de proveer información a sus clientes o partners sobre extranets o de Internet. Reporting Services aísla los reportes de clientes desde la complejidad de los data sources, mientras provee personalización e interactividad. La arquitectura abierta y extensible de reporting services ofrece una plataforma completa, basada en servidor, para creación administración y entrega tradicional o interactiva de reportes. Al mismo tiempo, el diseño modular y el alcance de las APIs (application programming interfaces) de reporting services posibilitan a los desarrolladores de software, proveedores de datos, y empresas, un manejo de informes integrado con sistemas legados o con aplicaciones third party.

Reporting Services ofrece una combinación única de atributos.

- Una completa plataforma basada en servidor para reporteo. Soporta el ciclo de vida completo, desde la creación hasta la distribución y administración continua de los reportes.

- Reporteo flexible y extensible. Soporta tanto reportes tradicionales como interactivos en numerosos formatos, con opciones de entrega extensibles. Puede ser integrado fácilmente en cualquier entorno o solución utilizando APIs abiertas o interfaces.
- Escalabilidad. La modularidad del producto, el diseño basado en servidor escala fácilmente para soportar entornos de alto volumen. Se puede crear una granja de servidores para reporting, con múltiples servidores de reportes, accediendo en el mismo core de reportes, sirviendo a miles de clientes basados en web.
- Integración con productos y herramientas Microsoft. Reporting Services se integra fácilmente a la familia de herramientas Microsoft, tales como Visual Studio y aplicaciones como Office y SharePoint Portal Server, sin requerir programación adicional ni adaptaciones.
- La creación, administración y entrega de informes de Reporting Services, combina los beneficios de un sistema de administración centralizada con la flexibilidad de aplicaciones web y de escritorio en demanda.
- Administración de reportes. Reporting Services incluye una herramienta web para administración de reportes así como la integración con la nueva Consola de SQL Server Management Studio. Los administradores pueden usar esta interfaz para organizar los reportes y Fuentes de datos, planificar ejecución y distribución de reportes, y rastrear los datos históricos. Una empresa o un ISV, pueden utilizar los APIs de los servicios web de Reporting Services, para escribir herramientas de administración personalizadas.
- Seguridad de reportes. Reporting Services implementa un flexible modelo de seguridad basado en un modelo de roles, para proteger los reportes y recursos de reporte. El producto incluye interfaces extensibles para integración de otros modelos de seguridad si es la decisión de la compañía.
- Distribución de informes. Se puede enviar los correos hacia un portal, mediante e-mail a los usuarios, o permitir a los usuarios utilizar el sitio web de report Server para que naveguen en la distribución jerárquica de reportes. La navegación, búsqueda y características de suscripción ayudan a los usuarios a localizar y correr los reportes que

ellos necesitan. Las suscripciones personalizadas les dejan seleccionar el formato de presentación de los reportes que ellos prefieran. La integración con los productos de Microsoft Business Intelligence hacen que Reporting Services se integre fácilmente con otros productos de la misma familia.

## **2.9. Office System**

Microsoft completa un verdadero conjunto de herramientas para inteligencia de negocios con las innovaciones presentadas en Visual Studio 2008, en lo referente a aplicaciones desarrolladas sobre Microsoft Office Excel, Microsoft Office Word, Microsoft Office PowerPoint. La versión 2008 al introducir la tecnología CLICKONCE para el manejo de certificados de seguridad y strongnames, ha superado los engorrosos problemas de publicación de versiones anteriores.

La utilidad de unir la funcionalidad de Excel con la programación en C# o VB, abre las puertas a un sin número de aplicaciones, entre las cuales claro, se encuentra la generación de reportes, sobre los cuales los usuarios pueden hacer uso de sus habilidades en las herramientas habituales como Excel o Word, para aprovechar la ventaja de acceder a datos sin necesidad de utilizar los mecanismos provistos en estas herramientas, haciendo que el acceso sea de cierto modo transparente.

Varias son las organizaciones que están incursionando en el mundo del desarrollo bajo tecnologías de Office, sin embargo, la especialización de otras herramientas, como reporting services del propio Microsoft, esa relegando la usabilidad que se le puede otorgar a Office System en el desarrollo de reportes. Y en gran medida, por las mismas características por las que se ha cuestionado a otras herramientas evaluadas en este documento.

## **2.10. Tendencias para la Generación de Informes Dinámicos**

En un sistema software complejo, con muchos y delicados procesos de cálculo y control, que arrojan datos que los usuarios de todos los niveles administrativos de una empresa deben analizar, evaluar, verificar y proyectar en busca de las mejores decisiones. Un reporte estático se convierte de manera incontrolable en una limitante y en el peor de los casos una barrera de obstáculos que hace que el trabajo se torne complejo e insostenible.

Esto sin lugar a dudas no coincide con la principal misión de un sistema informático, que es facilitar el trabajo del activo humano.

La tendencia actual de los sistemas software es proporcionar al usuario la libertad de acción sobre una aplicación, lógicamente enmarcado dentro de las capacidades mismas del software. En reportes, un usuario se siente fascinado por la idea de generar la información que domina de manera fácil y que cumpla con sus requerimientos. Por esta razón, las últimas generaciones de herramientas para elaboración y presentación de informes, han tenido que incluir medios para generación de reportes ad hoc. Sin embargo, es difícil determinar el equilibrio clave entre facilidad de uso y dominio en la generación de informes. En el presente trabajo, uno de los principales objetivos es establecer dicho equilibrio.

El dinamismo en la generación de reportes informáticos se queda enmarcado dentro de una frase Ad Hoc Quering.

#### **2.10.1. Ad Hoc Quering**

Muchos sistemas de software tienen una base de datos que puede ser accedida por solo un número limitado de usuarios, consultas y reportes. Típicamente, estos están disponibles por medio de alguna clase de menú y tienen que ser cuidadosamente diseñados, pre-programados y optimizados para un mejor rendimiento por programadores expertos.

En contraste, los sistemas "ad hoc reporting", permiten a los usuarios crear consultas personalizadas y específicas por sí mismos. Generalmente lo pueden hacer por medio de un sistema basado en una interfaz de usuario GUI amigable, sin la necesidad de profundizar en el conocimiento de SQL o modelos de base de datos que un programador debe tener.

Dado que un reporte tiene el potencial para degradar severamente el rendimiento de un sistema, en esas ocasiones proveer una sola copia regularmente refrescada de la vida de la base de datos, dependiendo del contexto como una copia puede ser referenciada como en datawarehouse las consultas como minería de datos.



## **2.11. Análisis comparativo entre SSRS 2008 y SAP Business Crystal Reports 2008**

### **2.11.1. Madurez del Producto**

Crystal Reports tiene más de 12 versiones y actualmente está en la versión 2008, el producto se ha enfocado por completo a las necesidades de elaboración de informes y ha madurado suficientemente con el paso de los años. Reporting Services en cambio, apenas ha sacado su tercera versión con SQL Server 2008, aunque este producto es relativamente nuevo, ha impactado en el mercado fuertemente, dado que Microsoft tiene la ventaja de realizar un análisis de las soluciones para elaboración de informes en el mercado implementando las características máspreciadas en SSRS. (Patra, 2008)

### **2.11.2. Licenciamiento**

Crystal Reports provee varias ediciones para diseño de reportes. Tres ediciones están embebidas en Visual Studio .NET sin costo adicional para desarrollo. Hay versiones stand alone independientes de Visual Studio y el paquete es vendido a una licencia por usuario de forma separada.

SSRS está incluido en la licencia de Microsoft SQL Server, sin embargo, por cuestiones de rendimiento se recomienda que el servidor de reporting services este en una instancia y servidor distintos, debido al alto consumo de recursos, esto hace necesario la adquisición de una licencia adicional SQL Server para mantener a Reporting Service en un entorno aislado. A pesar de ello, los costos resultas mucho más asequibles que la adquisición de licencias por usuario de Crystal.

### **2.11.3. Formato del Archivo de Reporte**

Crystal Reports está almacenado en formato binario. El modelo de objetos de Crystal puede ser usado programáticamente accediendo y manipulando los componentes de reporte.

Los reportes creados mediante SSRS son almacenados como texto plano XML. Este puede ser abierto en un editor de texto y los cambios pueden hacerse sin necesidad del diseñador

de Visual Studio .Net. Podría asumirse que es una opción bastante conveniente, pero manipular el archivo XML puede resultar bastante complejo si es que no se tiene un conocimiento basto de la nomenclatura de los elementos. Pero para ello SSRS presenta Report Builder, herramienta gratuita que permite manipular reportes en ambiente desconectado.

#### **2.11.4. Acceso a Datos**

Crystal Reports tiene mayor variedad de drivers de enlace hacia distintas fuentes de datos, sin embargo, SSRS también soporta los proveedores más habituales como ODBC, ADO, OLAP, XML, entre otros. La ventaja de Crystal en este aspecto es pocas veces útil en la practicidad. La diferencia está en que en Crystal se nota un cuello de botella al momento de enlazar en tiempo de ejecución los reportes, ya que para realizar esta acción se utilizan librerías de COM+, con Interop, con la conocida complejidad que agrega a la aplicación. Esta es una de las razones por las que Crystal Reports perjudica al rendimiento de la aplicación.

En cuanto a acceso a datos de ERPs o CRMs también lleva algo de ventaja Crystal Reports, pero los drivers para acceder a esos orígenes de datos normalmente se adquieren aparte.

#### **2.11.5. Data Binding**

Este es un aspecto importante en el desarrollo de reportes, Crystal Reports permite una única fuente de datos por reporte, si los datos son obtenidos desde múltiples tablas de la base se requiere hacer un join de las tablas para obtener un único conjunto de datos. Para incluir múltiples fuentes de datos es necesario incluir el mismo número de subreportes en el reporte principal. Aquí hay otra limitación para Crystal, solo se acepta un nivel de profundidad en sub reportes.

SSRS es flexible en este sentido, permite cualquier número de fuentes de datos por reporte. Esto facilita el diseño de reportes con secciones que despliegan datos desde

diferentes fuentes de datos. SSRS tampoco presenta limitación en el anidamiento de sub reportes, soporta más de 20 niveles de profundidad.

#### **2.11.6. Diseñador de reportes**

Crystal ofrece una funcionalidad inexistente en Reporting Services, la posibilidad de editar el informe en la ventana de vista previa. Tiene un diseñador que se agrega al IDE de Visual Studio .Net al momento de instalarse, es intuitivo y dispone de un gran número de wizards llamados **Utilidad** + “Expert”. Reporting Services ofrece un asistente para la creación de informes sin embargo, las tareas más comunes deben realizarse utilizando los cuadros de dialogo que requieren un tanto de destreza hasta asociarse con la herramienta. No obstante, SSRS 2008, ofrece mucha mayor facilidad operativa que sus antecesores. No existiendo mayor problema para el uso de un desarrollador. Adicionalmente, la herramienta Report Builder, que está diseñada para usuarios no técnicos, ofrece toda la usabilidad requerida por un end user.

Crystal Reports está totalmente condicionado al tamaño del papel para el diseño del informe, la longitud de las páginas, márgenes y espacios dependen totalmente del papel elegido en las opciones de impresión. En SSRS el diseño es mucho más flexible, lo cual trae sus consecuentes ventajas para diseño de aplicaciones Web, que trae un costo adicional mínimo al momento de diseñar para papel.

#### **2.11.7. Características de Diseño de Reportes**

- **Capacidad WYSIWYG.** Dado el grado de madurez de Crystal Reports, la interfaz de diseño presenta una clara imagen del reporte en modo de ejecución. Las anteriores versiones SSRS presentaban problemas de desplazamiento en vista preliminar con respecto a la vista de diseño debido a problemas de alineación y distribución de objetos, pero ha implementado mejoras en la interfaz que hacen que todos los elementos estén alineados y se tiene un buen nivel de WYSIWYG.
- **Estructura del Reporte.** En Crystal Reports se utiliza una estrategia basada en secciones, hay secciones predefinidas en un orden específico (por ejemplo header /

page footer / group header / footer / body etc.) y los diseñadores personalizan cada sección con el aspecto decidido. SSRS utiliza una estrategia diferente, es posible arrastrar y soltar, y personalizar los elementos dentro de la interfaz de diseño. Esto provee mayor flexibilidad en la definición de la estructura de los reportes.

- **Fórmulas y Cálculos.** Crystal Reports provee un amplio rango de fórmulas predefinidas que están listas para ser usadas en los reportes. SSRS provee todas las funciones básicas, pero no iguala a Crystal Reports en términos de la extensión de funciones. Sin embargo, hay una premisa que eventualmente puede cambiar la balanza a favor de SSRS y es que todas las funciones de VB pueden utilizarse directamente en la escritura de Expresiones de SSRS, es decir que para un desarrollador, se presenta una amplia gama de posibilidades.
- **Agrupación/Subtotales.** “En Crystal Reports solo puede existir una jerarquía de grupos. Es decir solo se pueden agrupar los datos de una manera (un grupo anidado dentro de otro y así sucesivamente) y esta agrupación define la estructura de todo el informe. En Reporting Services existen una serie de elementos de diseño, por ejemplo tablas y matrices, que permiten, cada una de ellas, definir sus propias opciones de agrupación, ordenación, filtrado, etc. Además es posible insertar unos elementos dentro de otros (por ejemplo anidar un gráfico o una tabla dentro de otra tabla). De modo que el diseño de un informe con una estructura algo compleja en Reporting Services se puede elaborar directamente, y de manera muy sencilla, mientras que en Crystal Reports exige la utilización de múltiples subinformes que, aparte de ser engorrosos, reducen el rendimiento de la generación del informe, ya que cada informe tiene que realizar su propia consulta al origen de datos. En reporting Services es muy sencillo reutilizar los datos de una consulta para mostrarlos bajo diversos puntos de vistas (filtrados, ordenaciones y agrupaciones diversas)” (Equipo Danysoft, 2006).
- **Extensibilidad.** SSRS provee opciones para escribir código personalizado o hacer llamadas a assemblies .Net desde el reporte.

- **Secciones para administración de flujo de datos.** Crystal Reports provee mejores características para manejar saltos de página, saltos de sección, espaciado de párrafos, etc. SSRS no hace un trabajo al mismo nivel. Estas características son altamente apreciadas en la generación de informes para banca.
- **Despliegue del Reporte en una Página Web.** La componente para visualizar reportes en web de Crystal Reports presenta algunas falencias en la presentación de las propiedades visuales de elementos como gráficas e imágenes, propiedades como transparencia y bordes se pierden una vez que el reporte es presentado en una página web. SSRS tiene igual comportamiento en una página web como en una aplicación Windows, dado que el visor es el mismo, y se especializa en reportes basados en servicios.
- **Texto Enriquecido.** Un punto débil de SSRS con respecto a Crystal es el hecho que no soporta campos con formato RTF.

#### **2.11.8. Formatos de Exportación**

Crystal Reports soporta alrededor de 16 formatos de salida, que incluyen claro, los formatos que SSRS soporta, HTML, MHTML, PDF, TIFF, Excel, Word. A pesar de la diferencia notoria en cantidad de formatos, SSRS soporta los más comunes, y adicionalmente, la exportación a Excel desde Reporting Services es muy superior, y mucho más fácil de controlar que en Crystal Reports.

#### **2.11.9. Manejo de Parámetros**

En ambas herramientas los parámetros pueden ser dinámicos, es decir, pueden mostrar una lista de valores posibles obtenida desde una consulta a base de datos y en cascada, los valores mostrados en una lista de parámetros dependen del valor proporcionado a un parámetro anterior. Pero esta funcionalidad está implementada en Reporting Services de una manera mucho más efectiva. En Crystal Reports todos los niveles de listas de la cascada de un parámetro se deben generar desde una única consulta a la base de datos, obligando a traer todas las combinaciones posibles de valores de los parámetros. Además,

según la documentación de Business Objects, esta consulta debe ser la propia del informe, y no una separada para los parámetros (un mensaje de error nos advierte de que puede haber problemas si tratamos con más de un origen de datos en Crystal Reports). Esta es una limitación enorme. En Reporting Services cada nivel de los parámetros en cascada proviene de una consulta diferente, que puede estar condicionada por los valores elegidos para parámetros anteriores, de modo que se obtienen del origen de datos los datos estrictamente necesarios para cumplimentar las listas: es decir, una vez que el usuario elige un valor para un parámetro se genera dinámicamente una consulta dependiente del valor de dicho parámetro para generar la nueva lista.

#### **2.11.10. Arquitectura**

Algunos expertos consideran que la desventaja en la arquitectura de SSRS es la dependencia hacia una base de datos. Crystal no depende de una base de datos, pero sí de la conectividad mediante COM+ e Interop lo cual tiene su repercusión en rendimiento.

#### **2.11.11. Costos**

Cada una de las licencias de Crystal, necesario por usuario, cuesta alrededor de \$700. Aunque SSRS implica en un sistema grande la adquisición de una licencia para SQL, no tiene restricción en el número de usuarios, por lo que a la larga, es mucho más conveniente que Crystal. Adicionalmente, si no se estima mucha carga en el sistema, Report Server puede convivir sin problemas con el motor de base de datos del negocio.

#### **2.11.12. Reportes AdHoc**

SSRS lanzó una nueva herramienta con la versión 2005, Report Builder, la misma que permite al usuario final crear informes a medida a través de una aplicación sencilla. Como se ha explicado en secciones anteriores 42. Para poder realizar estos informes es necesario que exista un servidor de informes y además se debe haber definido previamente un modelo de datos de donde obtendrán su información los informes. En Business Objects, una funcionalidad similar solo está disponible en los productos de su familia Enterprise, con los costos adicionales que ello implica.

#### **2.11.13. Tablix**

SSRS incluye una nueva región de datos tablix que combina una tabla con una matriz, más que un control en si Tablix es una nueva funcionalidad para crear reportes más robustos, donde se pueden anidar grupos a nivel de filas y columnas. Las cabeceras de filas y columnas pueden ser opcionales. Ninguna funcionalidad de Crystal es comparable con Tablix.

#### **2.11.14. Gráficas**

Antes de la versión SSRS 2008, Crystal Reports ejercía una clara ventaja en los controles para gráficas y medidores, sin embargo en esta última versión Reporting Services adquirió el producto DUNDAS, para la visualización de datos, los gráficos son fáciles de usar.

#### **2.11.15. Tipos de Reportes**

Aunque las dos herramientas soportan todos los tipos de informes, Crystal solo soporta un nivel de subreportes.

En SSRS las funcionalidades interactivas como drillthrough reports están disponibles en formatos de exportación basados en Web, pero no en todos los formatos basados en imágenes.

#### **2.11.16. Conclusión**

Ambas herramientas proporcionan características bastante amplias para construir reportes en una aplicación, mientras Crystal Reports toma crédito por permanecer en el campo varios años, SSRS provee características entusiasmantes en la plataforma .Net. Para tomar una decisión es necesario identificar los requerimientos primarios y entonces evaluar los productos y decidir cuál es el más apropiado.

### **2.12. Cuadro comparativo de SAP Business Objects vs SQL Server Reporting Services**

En base al análisis realizado en capítulos anteriores de las plataformas existentes en el mercado para elaboración de informes, para el objeto del presente estudio, se estableció un

análisis más detallado para dos de de las plataformas más robustas, siendo éstas SAP Business Crystal Reports y Microsoft SQL Server Reporting Services.

El siguiente cuadro muestra un resumen de los atributos de cada plataforma, al mismo tiempo que realiza una asociación de la evaluación cualitativa, con un factor de peso para que éste pueda ser utilizado posteriormente<sup>1</sup>.

La cuantificación de los parametros indicadores no está basada únicamente en el número de propiedades, herramientas, drivers, plataformas, o formatos soportados. Más bien es una conjugación de elementos como calidad, soporte de características comúnmente requeridas en el mercado, aprovechamiento real de una característica dada.

En cada una de las características, se cita los factores más relevantes a tomar en cuenta en la plataforma, que determinan las fortalezas y debilidades en una categoría dada.

---

<sup>1</sup> Los datos de los cuadros siguientes están basados en entrevistas, experimentación y datos publicados en (Evelson, 2008),



**Tabla II-29.** Cuadro comparativo de características CR vs SSRS

Indicador	SAP Business Objects		SQL Server Reporting Services	
<b>Producto</b>				
<b>Madurez del Producto</b>	Más de 12 versiones	9	3 versiones	8.5
<b>Licenciamiento</b>	Por usuario, los kits adicionales así como el Central Management Server	7	Incluido en SQL Server 2008	9
<b>Costos</b>	Alta inversión en una solución con varios clientes / mediana inversión en un sistema pequeño	7	Baja inversión en una solución con varios clientes / en una sistema pequeño si no se presenta excesiva carga al servidor BD, la inversión en nula, caso contrario es mediana.	9
<b>Promedio</b>		<b>7.67</b>		<b>8.83</b>
<b>Arquitectura</b>				
<b>Principal dependencia</b>	COM+, Interop	8	Base de Datos	9
<b>Plataformas soportadas</b>	Todos los componentes pueden ser instalados bajo Microsoft Windows (Java Reporting Component) <ul style="list-style-type: none"> <li>• Linux Red Hat Enterprise 3</li> <li>• Solaris 8 o superior</li> <li>• AIX 5.1 o superior</li> <li>• HP-UX 11i v1</li> </ul>	10	Solamente en la plataforma Windows, se requiere MS SQL Server 2000 o superior	7
<b>Soporte multi capa</b>	Completamente Escalable	9.5	Herramientas, ePortFolio, Crystal Management Console, Wizards y Crystal Configurator Manager	9
<b>Cliente</b>	Herramientas, ePortFolio, Crystal Management Console, Wizards y Crystal Configurator Manager	9	Report Manager, Report Designer, Reporting Services ConfiguratorManager, Report Builder	10
<b>Servidor</b>	Módulos CE: APS(manageja seguridad, objetos Crystal y servidores), Cache Server(sirve páginas procesadas previamente) y el Web Component Server (servidor web y add-on para procesamiento de reportes y distribución)	9	Report Server	10
<b>Procesamiento de reportes</b>	Crystal Job Server, Crystal Page Server: divide reportes previamente procesados en sus páginas, con el fin de preservar el ancho de banda. Integrado con IIS	9	Report Server	9
<b>Soporte de Clúster</b>	Cualquier módulo puede ser instalado en un server clúster sobre Windows. Esta característica tiene utilidad especialmente para APS y módulos Job Server. Disponible in Crystal Reports Server	9	Soporta arquitectura scaled out	10
<b>Balance de Carga</b>	Uno de los módulos APS funciona para el procesamiento de balance de carga entre todos los servidores Job disponibles en la LAN. El balance permite priorizar ciertos usuarios y reportes. Disponible en Crystal Reports Server a través de Job Server	9	El report server puede ser configurado en una granja de servidores (se incurre en costos adicionales para licencias SQL), y se puede utilizar el Resource Governor para administrar el balance de carga.	9

<b>Cache</b>	CE incluye un servidor de cache que convierte y muestra las páginas reprocesadas por Crystal Page Server. Puede mostrar en formato EPF(formato de página encapsulado) o HTML. Disponible en Crystal Reports Server mediante Cache Server	9	RS permite colocar el reporte y sus datos en cache, mediante la configuración de una opción del reporte para el tiempo solicitud en la generación. De este modo el primer usuario lanza el proceso de generación y los demás recuperan desde la cache.	10
<b>Archivos Log</b>	Un archivo log para cada módulo que es parte de CE con el objeto de crear una traza del módulo. Mediante el monitor del sistema, Windows Event Log, Windows Performance Log	8	Report Server execution log. La información acerca de los reportes específicos incluyendo cuándo y por quien fue ejecutado, destino y forma de presentación. Trace log. Información para aplicaciones en depuración. Windows Event Log, Windows Performance Log	10
<b>ERM/CRM</b>	A través de "Solution Kits", se puede acceder a ERP/CRM como SAP, Baan, Siebel y People Soft. Integrando seguridad/autenticación ERP/CRM . Las licencias se venden por separado.	9	No se incluye de forma directa. Pero es posible desarrollar Extensiones de Procesamiento de Datos para fuentes de datos específicas, como .NET Data Providers	8.5
<b>Soporte Servidor Web</b>	IIS, Apache/Tomcat (Java connectivity)	9	IIS	8
		<b>8.96</b>		<b>9.13</b>

#### Estructura del Reporte

<b>Distribución de los elementos</b>	Por secciones.	8	Existe libertad para la ubicación de los objetos.	9
<b>Formato del Archivo del Reporte</b>	Formato binario, solo se puede editar mediante las herramientas de diseño.	8	Formato XML, se puede crear/modificar con cualquier editor de texto.	10
		<b>8</b>		<b>9.5</b>

#### Servidor de Reportes

<b>Publicación</b>	Crystal Decisions presenta dos herramientas basadas en web : ePortfolio, una herramienta cliente, y Crystal Management Console (CMC), una herramienta administrativa. ePortfolio permite acceder, ver, imprimir, exportar, programar y administrar los reportes publicados. CMC permite administrar y configurar el sistema, reportes, grupos, usuarios, etc.	8	RS incluye un único entorno web para administrar y ver reports. La publicación es manejada por el Report Manager o por el report designer embebido en .Net. En modo integrado con SharePoint la administración se hace desde esta herramienta. También es posible utilizar el comando rs en consola.	10
<b>Publicación: tipos de objetos</b>	Reportes (Crystal Reports), cubos (Crystal Analysis), archivos (pdf, xls, doc, ...), aplicaciones (exe, bat, ...) y paquetes de reportes	9	Reportes (Archivos RDL), con sus fuentes de datos compartidas. Cualquier otro tipo de archivo puede mantenerse con formato binario, llamado recurso. Al momento de la publicación, Report Server graba una copia del archivo original, y no se mantiene ningún vínculo con el archivo original.	9
<b>Programación de reportes: General</b>	Bajo demanda, se mantienen varias instancias (ejecuciones históricas) del mismo reporte.	8	Las suscripciones a reportes se ejecutan bajo demanda acorde a la programación.	9.5
<b>Programación de tiempo</b>	Ejecución en un momento particular, recursivamente, o mediante plantillas.	9	Generación de Snapshot, también llamado instancias en CR, planificación en un tiempo dado, recursivamente o mediante programaciones compartidas.	9.5
<b>Calendario de programación</b>	Por plantillas, llamadas calendarios, los cuales pueden ser compartidos por diferentes objetos. Se puede definir plantillas por usuarios o grupos.	9	Conocidos como Shared Schedules, permiten la definición de programaciones que pueden ser usados por diferentes reportes.	9

<b>Formatos de exportación</b>	Más de 16 formatos <ul style="list-style-type: none"> <li>• HTML 3.2/4.0</li> <li>• PDF</li> <li>• Excel</li> <li>• XML</li> <li>• Word</li> <li>• RTF</li> <li>• Report Definition</li> <li>• CSV</li> <li>• Text Files (TAB y comma delimited)</li> <li>• ODBC</li> </ul>	10	Soporta los formatos más habituales <ul style="list-style-type: none"> <li>• HTML con Componentes Office Web</li> <li>• PDF</li> <li>• Excel</li> <li>• Archivos XML con datos de reporte</li> <li>• Web archive</li> <li>• TIFF</li> <li>• CSV (comma delimited)</li> <li>• Custom Rendering vía .NET assemblies</li> </ul>	9
<b>Canales de entrega</b>	Los reportes ejecutados pueden ser entregados mediante varios canales: MAPI, archivos, FTP, etc.	9	Por medio del uso de las extensiones de distribución (extensión de entrega e-mail y extensión de entrega por archivos compartidos). Es posible desarrollar extensiones de entrega personalizadas	9.5
<b>Selección de impresora</b>	Es posible especificar que instancia del reporte generado se imprimirá mediante una impresora en particular. Esa impresora puede ser configurada en el servidor CR o localmente. Se puede cambiar la configuración de la impresora.	9.5	No hay una extensión de distribución que envíe un reporte ejecutado a una impresora. Sin embargo es posible desarrollar una extensión de distribución con este propósito.	9
<b>Subscripciones basadas en datos</b>	No disponible	0	Es posible crear suscripciones dinámicas como resultado de una consulta. La extensión de entrega utilizada para suplir el reporte dependerá de los valores reportados por una consulta a la base de datos.	9
<b>Alertas</b>	CE reconoce alertas definidas en el reporte y las administra como un objeto independiente. Las alertas permiten resaltar datos en el reporte ejecutado mediante condiciones definidas. Cuando se abre un reporte ejecutado, este despliega alertas desencadenadas y los datos asociadas a ellas.	9	No disponible	0
<b>Eventos</b>	Hay tres tipos de eventos: personalizados, eventos programados y archivos. El propósito es concatenar pasos programados dependiente del resultado de un paso dado.	8	No disponible	0
<b>Historial de reportes</b>	Mediante planificación de instancias	8	Mediante snapshots, instancias de reportes que contienen el diseño y datos provistos para una fuente de datos específica.	9
<b>Asistente de publicación</b>	En CR, con una conexión directa a APS en CE, mediante el asistente de publicación de Crystal o CMC (Crystal Management Console). También se puede publicar cualquier reporte programáticamente utilizando el modelo de objetos (SDK)	9	La publicación de reportes puede hacerse desde Visual Studio .NET o desde Report Manager, mediante línea de comandos o programáticamente.	9.5
<b>Reportes enlazados</b>	No disponible	0	Un reporte enlazado es como un acceso directo a un reporte existente que comparte algunas propiedades compartidas con el informe original y tiene propiedades propias. El reporte original provee información de la conexión, la consulta utilizada para obtener los datos y el diseño del reporte. El reporte	9.5

			enlazado tiene nombre, descripción, ubicación, parámetros, ejecución del reporte, historia, seguridad y suscripción propios.	
		7.54		7.96
<b>Herramientas</b>				
<b>Diseñador de reportes</b>	Crystal Reports. Edición del reporte en vista previa. Condicionado al tamaño de papel.	8	Libertad en la distribución de impresión.	9
<b>Servidor de Reportes</b>	Con CE o Applications Server Report (RAS). Ambas herramientas publican en la web. RAS tiene menor funcionalidad.	8	Report Server	9
<b>Administrador de Conexiones</b>	Mediante "Business Views" se puede definir reglas del negocio que pueden ser incorporadas en tiempo de diseño.	9	Mediante el Modelo de Datos se puede seleccionar datos y establecer esquemas adicionales para que estén disponibles al usuario final	9.5
<b>Publicación de reportes</b>	En aplicaciones Windows mediante (Crystal Reports, Crystal Publishing Wizard), en aplicaciones Web (Crystal Management Console)	9	Mediante report designer en desarrollo, o report designer producción, también se puede utilizar el servicio Windows vía consola de comandos. En modo integrado con SharePoint, sustituye al ReportManager.	9.5
<b>Herramientas Cliente</b>	Report Designer (CR)	9	Business Intelligence Development Studio	9
<b>Herramientas Servidor</b>	Disponible en Crystal Reports Server Edition	9	Report Manager (IE)	9.5
		8.67		9.25
<b>Integración</b>				
<b>Integración con Sharepoint</b>	No disponible	0	Disponible con ciertas restricciones	8.5
<b>Extensibilidad</b>	No disponible	7	Escritura de código personalizable o llamadas a assemblies.	9
<b>.Net Framework</b>	El .Net SDK ofrece funcionalidad CE para personalizar herramientas cliente (ePortfolio o Crystal Management Console)	8	La personalización de Reporting Services puede realizarse mediante Web Services (ellos exponen todo el servidor de reportes), solicitudes URL, Proveedor WMI (control de acceso, personalización y administración), creación y personalización de extensiones mediante componentes RS, distribución, procesamiento de datos, etc., y lenguaje de programación RDL.	9.5
<b>Microsoft Office integration</b>	Asistente para crear reportes desde Microsoft Excel y Microsoft Access. Soporta Excel/Access 97 o superior. También se puede incluir objetos Crystal en documentos Office XP o superior con Smart Tags	8	RS dispone de una fuerte integración con una gran variedad de productos Microsoft, como Internet Explorer 5.5 o superior, MS Office 2000 o superior, MS SharePoint Portal Server y MS SQL Server Analysis Services 2000 o superior. Report Designer puede importar reportes que fueron creados en una base de datos MS Access (.mdb) o archivos de proyecto (.adp). El diseñador de reportes convierte el informe a un archivo RDL. La exportación a Office Web Components (OWC) facilita la interacción con el cliente. Algunos gráficos pueden ser exportados a los controles gráficos de MS Office y los controles de matrices pueden ser convertidos en controles PivotTable.	9.5
<b>Integración con Visual Studio.NET</b>	Disponible	9	Disponible	9

<b>Soporte SDK</b>	.NET, COM, Java, Crystal Reports API	9.5	.NET Support vía Web Services	9
<b>Soporte solo cliente</b>	Disponibles, todos los informes pueden ser vistos en una máquina cliente, sin que sea necesaria una conexión al servidor.	9	Requiere conexión al report server a través de HTTP o SOAP para procesar y crear el informe.	7
		<b>7.21</b>		<b>8.79</b>
<b>Manejo de Seguridad</b>				
<b>Autenticación</b>	Seguridad integrada vía IIS y Windows (Solamente CR). Soporta varios tipos de autenticación: Seguridad CE, NT, LDAP, Active Directory y Seguridad Lotus . Es posible añadir varios tipos de autenticación mediante kits respectivos como Siebel o autenticación SAP	9	Seguridad integrada vía IIS y Windows para conectarse al Report Server. Administración mediante Internet Information Services (IIS) y seguridad Windows. Los usuarios que se conectan al report server deben tener una cuenta windows válida.	9
<b>Seguridad de los Objetos</b>	Disponible en Crystal Reports Server Edition. Hereda la seguridad de las carpetas y archivos, a través de usuarios Windows, grupos de usuarios o roles de usuarios CE. Los derechos se pueden asignar a objetos en términos individuales.	9	Seguridad basada en Windows, sobre usuarios Windows, grupos de usuarios, y roles de usuarios RS. Los permisos pueden asignarse a los objetos en términos individuales.	9
<b>Programación de tiempo</b>	Una vez y periódicos a través de Crystal Reports Server Edition	9	Una vez y periódicos	9
<b>Entrega de informes</b>	Disponible en Crystal Reports Server Edition. (MAPI, SMTP, FTP, archivos)	9	Correo electrónico SMTP como un enlace al Report Server o como un archivo exportado adjunto; Entrega de archivos compartidos	9
<b>Seguridad de acceso a datos</b>	Mediante Business Views se puede dar seguridad al acceso a datos. La seguridad se aplica a objetos, columnas y filas.	9	La seguridad de los datos depende de la fuente de datos, las credenciales requeridas por la fuente de datos son necesarias siempre. Se puede acceder a los datos mediante fuentes de datos compartidos o conexiones definidas en el archivo RDL	9
		<b>9</b>		<b>9</b>
<b>Visor de reportes</b>				
<b>WYSIWYG</b>	Disponible	9	Facilita la alineación mediante líneas de referencia entre objetos.	9
<b>Thick-client</b>	Control personalizado embebido	8	Control personalizado embebido	8
<b>Visor Web</b>	No conserva propiedades visuales como transparencia y efectos de bordes. Visor HTML/DHTML ActiveX Control Java	8.5	Visor común para la presentación de informes. HTML 4.0 para IE 5.5 o superior HTML 3.2 para IE 5.1 o inferior, o navegadores Netscape/Mozilla	9.5
<b>Capacidades de exportación</b>	El procesador del servidor de reporte puede exportar a múltiples formatos	9	Exporta a los siguientes formatos (utilizando Extensiones de Renderización incluidas en RS): DHTML, xls, text, xml, image y pdf. También es posible desarrollar extensiones personalizadas de renderización.	9
<b>Impresión local</b>	Disponible	9	Disponible	9
		<b>8.7</b>		<b>8.9</b>
<b>Conectividad a Bases de Datos</b>				
<b>Fuentes de Datos Disponibles</b>	Conectividad nativa a: • Access (DAO) • ADO.NET (XML)	10	Extensiones de procesamiento de datos a través de ADO.NET • SQL Server 2008 • SQL Server 2005	9.5

	<ul style="list-style-type: none"> <li>• COM Connectivity</li> <li>• Database Files (Excel, Paradox, Btrieve, XML, Access files, etc.)</li> <li>• Exchange 5.5 Message Tracking Log</li> <li>• Exchange Message Tracking Log</li> <li>• Field Definition Files</li> <li>• File System Data (Archivos de texto)</li> <li>• Java Beans</li> <li>• JDBC</li> <li>• Legacy Exchange</li> <li>• Mailbox Admin</li> <li>• IIS/Proxy Log Files</li> <li>• NT Archived Event Log</li> <li>• NT Current Event Log</li> <li>• ODBC (RDO)</li> <li>• OLEDB (RDO)</li> <li>• Outlook</li> <li>• Outlook/Exchange</li> <li>• Public Folders</li> <li>• xBase</li> <li>• XML</li> </ul>		<ul style="list-style-type: none"> <li>• SQL Server 2000</li> <li>• SQL Server 7.0</li> <li>• SQL Server 2005 Analysis Services</li> <li>• SQL Server 2000 Analysis Services</li> <li>• Oracle (debe haber instalado un cliente Oracle en el Report Server)</li> <li>• ODBC Data Sources</li> <li>• OLEDB Data Sources</li> <li>• XML</li> <li>• Hyperion Essbase</li> <li>• SAP NetWeaver BI</li> <li>• TERADATA</li> <li>• Extensible para acceder a fuentes de datos adicionales como Event Logs, Exchange, etc..</li> </ul> <p>Otras fuentes de datos pueden requerir drivers separados o utilitarios que deben ser instalados en el Report Server</p>	
<b>Datos Compartidos</b>	Para crear fuentes de datos compartidas entre reportes se debe utilizar la funcionalidad "Business View".	8	Las fuentes de datos pueden ser compartidas entre múltiples reportes. No se graban como parte de un informe y son administradas por el servidor RS.	9.5
<b>Data Binding</b>	No soporta múltiples fuentes de datos. Para hacerlo se requiere sub reportes	6	Número de fuentes de datos ilimitado	9.5
<b>Comandos SQL</b>	Las sentencias SQL pueden crear una tabla virtual y usarla como fuente de datos del reporte. Estos comandos pueden mantenerse en el repositorio de Crystal para compartirlos con otros usuarios.	9	No disponible. Sin embargo, los comandos SQL no son necesarios en RS porque existe la posibilidad de crear y modificar la consulta que llena el conjunto de datos del informe.	8.5
<b>Conectividad abierta</b>	Se conecta a cualquier fuente de datos definida en el diseño del reporte (ODBC, OLEDB, .Net, COM, JavaBeans, sus propias fuentes de datos, etc.) Se incluye drivers datadirect.	10	RS se conecta a las fuentes de datos mediante las extensiones de procesamiento de datos. Utilizando proveedores .Net. (OLE DB .Net Data provider, ODBC .Net Data Provider). No se incluyen extensiones de procesamiento de datos para bases de datos de PC (Access, dBase, Paradox).	9
<b>Proveedores de datos COM personalizados</b>	Solo está disponible en Advanced Edition	8	Extensiones de procesamiento de datos personalizables. Posibilidad de desarrollar extensiones de procesamiento de datos para fuentes de datos específicas como .NET Data Provider.	9.5
<b>Conjuntos de datos ADO.NET (XML)</b>	Disponible en ediciones Dev/Adv.	8	Disponible en la versión 2008.	9
<b>JavaBeans</b>	Se incluye conectividad a Java Beans connectivity para crear reportes desde elementos Java Bean (cualquier clase Java compilada que incluya un método	8.5	No disponible	0

	público que retorne un java.sql.ResultSet). Disponible en Advanced edition.			
<b>Motores OLAP</b>	MS OLAP, IBM DB2, Holos, Essbase	9.5	Extensión de procesamiento de datos para SQL Server Analysis Services	9
<b>Conexiones definidas en el servidor</b>	Disponible desde CE versión 10. Business Views definidas previamente para hacer más fácil el aprovechamiento de datos empresariales.	8	No disponible	0
<b>Conexiones nativas</b>	Informix, Oracle, SQL Server, Lotus Domino, Local File System, Web Log Files, Microsoft IIS/Proxy Log File, Exchange/Outlook y NT Event Log	9.5	Conexiones nativas con SQL Server .NET Data Provider a SQL Server 7, SQL Server 2000, SQL Server 2005, SQL Server 2008 y SQL Server Analysis Services; con Oracle .NET Data Provider y Oracle client con Oracle 8.1.7 y superior	9
<b>XML</b>	Disponible	9	Disponible	9
<b>Opciones de Conectividad</b>	Diferentes conexiones son administradas por el motor de reportes, ofreciendo las siguientes posibilidades. 1. Múltiples reportes puede compartir la misma conexión a la base de datos. 2. Múltiples conexiones pueden pertenecer simultáneamente a un mismo reporte. La información de conexión es almacenada siempre dentro del reporte.	9	Las conexiones son administradas por el proveedor de datos .Net Data Provider, de este modo, el pool de conexiones y otras responsabilidades son asumidas por este.	9
<b>Editor de consultas</b>	No se incluye. La única herramienta para crear y modificar consultas son los comandos SQL.	5	incluye el editor de consultas de SQL Server Enterprise Manager	9.5
		<b>8.39</b>		<b>7.86</b>

**Características para construcción de reportes**

<b>Asistentes</b>	Disponible	9.5	Disponible	9
<b>Asistente para la creación de un reporte en tiempo de ejecución</b>	CRWizard embebido en el IDE de Visual Studio .Net o en herramientas office	9	Asistente integrado con Visual Studio .Net, como parte de Business Intelligence.	7
<b>Objetos embebidos</b>	Se puede incorporar el diseñador CR dentro de la aplicación como un objeto embebido.	7	No disponible	0
<b>Tiempo de creación de un reporte en ejecución</b>	No se puede crear un reporte en tiempo de ejecución, solo modificarlo, el tiempo requerido es alto	6	Se puede optimizar el tiempo de creación utilizando las ventajas de LINQ. El tiempo de creación es medio.	8
		<b>7.88</b>		<b>6</b>

**Características para desarrollo de reportes**

<b>Desarrollo del reporte dentro de la herramienta de desarrollo</b>	Crystal Reports ActiveX Designer	9	RS - Report Designer	9
<b>Desarrollo de reportes AdHoc</b>	API para creación de reportes incluida en el CR SDK(.Net, COM y versiones JAVA)	8	Creación de un archivo XML con la extensión "rdl" siguiendo la especificación RDL.	9
		<b>8.5</b>		<b>9</b>

**Tipos de reportes**

<b>Reportes Drill Down</b>	Disponible soporta n niveles de profundidad	9	Disponible soporta n niveles de profundidad	9
<b>Subreportes</b>	Disponible solo soporta 1 subreporte por reporte principal.	8	Disponible, soporta n niveles de profundidad de sub reports.	9.5
<b>Reportes Cross-Tab</b>	Disponible	9	Disponible	9
<b>Reportes Olap</b>	Disponible	9	Disponible	9
<b>Reportes Multi-Columnas</b>	Disponible en la sección de detalle de reportes	9	Disponible en la sección de detalle de reportes	9
<b>Labels</b>	Estándar y definidos por el usuario	9	Únicamente definidos por el usuario	8
<b>Reportes Ad Hoc</b>	Disponible únicamente en la familia Enterprise, basado en plantillas	8	Disponible, basada en modelos de datos	9.5
		<b>8.71</b>		<b>9</b>

**Ordenamiento y Agrupación**

<b>Grupos Jerárquicos, Top-N, Bottom-N, Top %, Bottom %:</b>	Agrupamiento basado en datos jerárquicos como administrador/empleado.	8	Disponible, con mayor flexibilidad	9.5
<b>Ordenamiento de grupos: Top-N, Bottom-N, Top %, Bottom %</b>	Disponible, incluye asistentes	9	Disponible	9
<b>Ordenamiento específico</b>	Permite priorizar la posición de los elementos con independencia del orden general. Por ejemplo una instrucción que coloque al país "Perú" en primer lugar que el resto de países con criterio de ordenamiento general (ascendente o descendente).	9.5	No disponible, pero se puede conseguir el mismo efecto a través de un procedimiento almacenado.	8
<b>Ordenamiento compuesto y condicional</b>	Disponible, pero no en todos los objetos	8	Disponible en todos los objetos	9.5
<b>Agrupamiento específico</b>	Permite la definición de grupos que no existen en la fuente de datos. Por ejemplo se puede definir con "Iberia" a los países "España y Portugal"	9	Se puede obtener esta funcionalidad mediante el uso de expresiones en el agrupamiento, o traerlo de forma directa del procedimiento almacenado.	8
<b>Agrupación/Subtotales</b>	Solo una jerarquía de grupos	8	Flexibilidad en la distribución de grupos.	9.5
		<b>8.58</b>		<b>8.92</b>

**Formato**

<b>Formato de Grupos</b>	Disponible	9	Disponible	9
<b>Formato de Sección</b>	Se puede aplicar formato compuesto o condicional a grupos, cabeceras y pies, áreas, secciones y subsecciones.	9	Hay solo tres secciones en el reporte. Cabecera de la página, cuerpo, y pie de página. En la sección Body es posible encontrar regiones de datos ilimitadas, con múltiples secciones que incluyen cabecera de página, n cabeceras de grupo, detalles, n pies de grupos, y pie de página. La creación de un reporte de múltiples secciones es más flexible que CR pero exige un poco mas de práctica. El encabezado y pie de página no admiten campos de datos.	9
<b>Quiebre de páginas condicional</b>	Se puede aplicar formato condicional a los saltos de página, a secciones y grupo. Aplica para cabeceras y pies.	9	No disponible	0



<b>Gráficos</b>	Tiene soporte para más de 80 tipos de archivos gráficos. Totalmente customizable.	9.5	Gráficos personalizables, los gráficos son provistos por Dundas Software (Dundas Chart for .NET). Para mejor funcionalidad Dundas Software proporciona versiones actualizadas en ( <a href="http://www.dundas.com">http://www.dundas.com</a> )	9.3
<b>Formato compuesto y condicional</b>	No aplica para todos los objetos. El formato puede basarse en el valor de un objeto o en un valor de un objeto externo.	8	Aplica para todos los objetos. Igual o más poderoso que CR. El formato condicional puede basarse en expresiones con Visual Basic .NET o con código personalizable y también por ensamblajes embebidos o referenciados.	9.5
<b>Formato de campos</b>	Se puede aplicar formato condicional sobre campos de la base, fórmulas, expresiones SQL, parámetros, totales y campos especiales.	8	Se puede aplicar formato condicional sobre cualquier objeto.	9.5
<b>Formato de Objetos</b>	Se puede aplicar formato condicional a gráficas, mapas, matrices, grillas OLAP, subreportes, objetos OLE, imágenes, líneas y figuras.	8	Se puede aplicar formato condicional sobre cualquier objeto.	9.5
<b>Asistente Highlighting</b>	Se puede asociar señales a modo de semáforo sobre cualquier ítem del reporte. Esta funcionalidad es útil para resaltar cualquier valor aplicando formato condicional sobre el mismo.	9	No se incluyen semáforos. Pero pueden ser emulados mediante expresiones de formato condicional.	9
<b>Plantillas</b>	Disponible	9	Disponible	9
<b>Administración del flujo de datos</b>	Buen manejo de espaciado entre párrafos, saltos de página y de sección.	9	Menor nivel que CR en el manejo de elementos de espaciado y saltos.	8.5
		<b>8.75</b>		<b>8.23</b>
<b>Objetos de reportes</b>				
<b>Grillas OLAP</b>	Limitado a 1 nivel. Puede compartir la conexión con el reporte principal. También puede obtener subreportes bajo demanda. Los subreportes se insertan en el reporte principal como objetos OLE.	9	No se limita a 1 nivel de profundidad. Los subreportes bajo demanda no están disponibles y su definición no está insertada dentro del reporte principal.	9
<b>Fórmulas</b>	Amplio rango de fórmulas y funciones. Sintaxis Crystal o Basic. Posibilidad de incorporar cualquier fórmula en cualquier objeto. Las fórmulas pueden ser reutilizadas localmente o en el repositorio del servidor. Están integradas en el reporte.	9	Fórmulas y funciones básicas. Reconoce funciones Visual Basic. Con .Net assemblies, código .Net code embebido, o expresiones Visual Basic .Net para formato condicional. Los campos calculados pueden ejecutar operaciones sobre el conjunto de datos en todos los casos en donde el motor de base de datos lo permita.	9
<b>Campos de Funciones Agregadas</b>	Herramienta gráfica para diseñar campos resumidos. Es posible crear campos resumidos con el uso de fórmulas. Manejo de porcentajes en cualquier grupo.	9	No hay una herramienta gráfica para crear campos resumidos. Es posible crear funciones agregadas con expresiones VB.NET, usando la función resumida RunningValue.	8.5
<b>Figuras, líneas</b>	No es posible aplicar formato condicional a líneas ni figuras.	8	Soporta (BMP, JPG, GIF, PNG, JPE). Puede almacenarlos de forma embebida o como parte del proyecto, en la base de datos o invocando a un Url. Totalmente personalizable, los rectángulos pueden contener otros objetos.	9.5
<b>Funciones personalizadas o definidas por el usuario</b>	Disponible mediante ActiveX DLL	8.5	Disponible, mediante .NET assemblies o código embebido.	9.5
<b>Hyperlinks</b>	Cualquier objeto de reporte puede ser enlazado a un archivo externo (rpt o archivo de sistema), a una dirección web o un e-mail. Los enlaces pueden ser	9	Existe funcionalidad como Jump to report, Jump to bookmark, Jump to URL. Las mismas que permiten redireccionar a una parte del reporte o a una dirección url a	8.5

	condicionales.		través de las propiedades del campo de forma similar a un hipervínculo.	
<b>Listas</b>	No disponible	0	Región de datos con estilo compuesto. Este control es un contenedor de otros controles, permitiendo colocarlos libremente en la sección body del reporte. Se puede incluir agrupación y filtrado, así como subsecciones.	9
<b>Grillas/Tablas/Tablix</b>	No disponible	0	Elementos para manejo de filas y columnas, que permiten manipular la información de forma similar al de una hoja de cálculo, lo que brinda familiaridad tanto a desarrolladores como usuarios finales.	9.5
<b>Matrices</b>	La información se presenta en filas y columnas, las mismas que pueden ser agrupadas en n niveles. Es posible aplicar plantillas.	8.5	Con el objeto matrix RS se tiene funcionalidad similar a los crosstabs de CR y tablas pivot, es muy flexible. Se puede añadir objetos (gráficos, imágenes, etc.) como parte de las celdas. Se puede expandir o contraer niveles de grupos.	9.5
<b>Mapas</b>	Incorpora mapeo en reportes con MapInfo (ActiveX )	8	No disponible	0
<b>Objetos OLE</b>	Los objetos OLE registrados en el sistema o provenientes de una base de datos (como un objeto image ). Su contenido es independiente del resto del reporte.	9	Controles ActiveX. Su contenido es independiente del resto del reporte.	9
<b>Expresiones SQL</b>	Las expresiones SQL son sentencias procesadas por el motor de base de datos, usando funciones y operadores de la base de datos. Estas expresiones están integradas, como columnas, en la consulta SQL generada por Crystal Reportes. La diferencia con las fórmulas es que estas son procesadas por el motor de Crystal Reports.	8	Las expresiones SQL no son necesarias como objetos especiales, debido a que el editor de consultas permite la edición de la sentencia SQL para añadir expresiones SQL en el dialecto SQL nativo de la base de datos.	9.5
<b>Regiones de Datos</b>	No disponible In CR, al arrastrar y soltar un campo de la base de datos a la sección de detalles en un diseñador, el motor de reportes imprime todos los registros retornados por la ejecución de la consulta. Los resultados, de forma similar a RS, pueden ser desplegados en varias secciones de detalle, subreportes y otras funcionalidades pero requieren usuarios avanzados con experiencia.	8	En RS, al arrastra un campo de la base de datos al diseñador el motor muestra solo el primer registro retornado por la consulta en tiempo de ejecución, para ver todos los registros se debe usar un control de tipo región de datos, como una lista o un grid. Los controles disponibles para este efecto son Tables, Matrix, Chart y List.	9.5
<b>Tablix</b>	No disponible	0	Disponible	10
<b>Gráficas</b>	Buen conjunto de objetos	9.5	Graficas Dundas	9.3
<b>Texto Enriquecido</b>	Disponible	9	No disponible	0
		<b>7.03</b>		<b>8.08</b>
<b>Parámetros</b>				
<b>Parámetros en cascada</b>	Se puede hacer personalizando una página CSP o mediante una interfaz de usuario. Permite que un parámetro de una lista afecte el tipo o número de parámetros de otra lista.	7	RS permite la creación de un grupo de parámetros donde la lista de valores para un parámetro depende del valor de otro parámetro. El filtrado de una lista de valores de parámetros basado en otro se denomina parámetros en cascada.	9.5
<b>Parámetros de múltiples fuentes de datos</b>	No disponible.	0	Disponible	9.5
<b>Manejo de Parámetros</b>	Parámetros dinámicos, y en cascada restringidos a misma consulta de datos. - Los parámetros de entrada se definen mediante un cuadro de diálogo en tiempo de diseño.	8.5	Parámetros dinámicos, y en cascada provenientes de cualquier fuente de datos. - Los parámetros de entrada se definen mediante un cuadro de diálogo en tiempo de diseño.	9,5

	<ul style="list-style-type: none"> <li>- Es posible definir valores por omisión o sugerirlos.</li> <li>- No está disponible parámetros dinámicos que puedan ser obtenidos con código personalizado.</li> <li>- Los parámetros requieren que el reporte este en ejecución desde CR o cuando se programa/ejecuta en CE.</li> <li>- Pueden ser referenciados por cualquier objeto del reporte.</li> </ul>		<ul style="list-style-type: none"> <li>- Es posible definir valores por omisión o sugerirlos.</li> <li>- Están disponibles parámetros dinámicos. Es decir que se puede seleccionar un valor para un parámetro desde una lista generada por una consulta contra la fuente de datos.</li> <li>- Los parámetros deben ser solicitados en tiempo de ejecución.</li> <li>- Pueden ser referenciados por cualquier objeto del reporte.</li> </ul>	
		<b>5.17</b>		<b>9.50</b>

**Tipos de campos de la base de datos**

<b>Campos básicos (varchar, integer,...)</b>	Total control	9	Total control	9
<b>Campos Blob(memo, image, ...)</b>	Total control desde la versión v10. Se puede acceder al contenido de un campo Memo desde una fórmula.	9	Total control	9
<b>Procedimientos Almacenados</b>	CR puede utilizar procedimientos almacenados.	9	La fuente de datos de un dataset puede ser un procedimiento almacenado.	9
<b>Vistas</b>	Son accesibles.	9	Son accesibles.	9
<b>Sinónimos</b>	Son accesibles.	9	Son accesibles mediante la edición de una consulta.	8
		<b>9</b>		<b>8.8</b>

**Explorador de Objetos**

<b>Explorador de Campos</b>	Muestra todos los objetos del reporte. (Campos de la base de datos, expresiones SQL, totales y campos especiales, metadatos, contadores de página, etc.)	9.5	Muestra todos los objetos del reporte	9
<b>Explorador de Reportes</b>	Muestra metadata de todos los objetos del reporte.	9	No hay una interfaz de usuario para acceder a esa información. Como un bypass, se puede mirar el código fuente XML del RDL.	8
<b>Explorador de Repositorios</b>	Muestra todos los objetos publicados en el repositorio. En la versión 10 o superior el repositorio es reemplazado por la funcionalidad Business Views.	9	No está disponible de forma integrada. Para manipular las carpetas y sus objetos es necesario utilizar el Report Manager.	8
		<b>9.17</b>		<b>8.33</b>

**Apertura para personalización**

<b>Funciones definidas por el usuario</b>	Funciones externas: Con la librería de funciones del usuario (User Functions Library UFL ) se puede llamar a las funciones definidas por el usuario (User Defined Functions UDF ) por cualquier entorno de desarrollo que soporte la creación de DLLs ActiveX. Funciones Internas: Las funciones se crean dentro del reporte. Pueden ser exportadas al Repositorio para un uso global.	9	Funciones externas: Referenciando a assemblies .NET. Funciones internas: Mediante la propiedad code de la página en lenguaje Visual Basic.	9.5
<b>Plantillas</b>	Template Expert, alrededor de 12 plantillas, y se da la opción de optar por templates externos, se puede aplicar la plantilla en cualquier momento.	9.5	6 plantillas, solo pueden ser aplicados al crear el reporte desde el diseñador de Visual Studio .Net. No puede cambiarse de template una vez que el reporte ha	8

			sido creado.	
<b>Personalización</b>	Soporta personalización en eventos, Thick-client, proveedores de datos COM, funciones.	8	Código personalizable, canales de distribución, Thick-client, proveedores de datos COM, funciones, gráficos, figuras y parámetros personalizables.	9.5
		<b>8.83</b>		<b>9</b>

**Tabla II-30.** Cuadro comparativo de capacidad de definición y creación de reportes CR vs SSRS

Indicador	SAP Business Objects	SQL Server Reporting Services
<b>Solución de reportes</b>		
Soporte técnico y documentación	9	8
Planificación de despliegue de nodos	7	9
Presupuesto de la solución de reportes	7	9
Promedio	7.67	8.67
<b>Administración y despliegue de la solución de reportes</b>		
Arquitectura solución de reportes	9	9
Administración centralizada	8	10
Detección de errores	9	10
Promedio	8.67	9.67
<b>Control del archivo del reporte</b>		
Libertad para ubicación de elementos del reporte	9	10
Manejo de Versiones	8	9.5
Promedio	8.5	9.75
<b>Funcionalidad aprovechable del servidor de reportes</b>		
Capacidad para herramientas de publicación independientes	8	9.5
Flexibilidad de los objetos que se pueden incluir en el informe	9	9
Planificación de distribución al usuario	8	9.5
Capacidad de administrar suscripciones y seguridad programáticamente	6	9
Administración de historial de reportes	8	9
Facilidad para la publicación	9	9.5
Capacidad para crear enlaces entre reportes	6	9.5
Promedio	7.71	9.29
<b>Facilidad de uso</b>		
Facilidad de diseño de reportes	9	9
Facilidad en la configuración del servidor	8.5	9
Facilidad en administración de acceso a datos	9	9.5
Facilidad de configurar mecanismos de distribución	9	9.5
Promedio	8.88	9.25
<b>Capacidad para complementar la solución de reportes con lenguajes o herramientas familiares.</b>		
Capacidad para incluir lenguajes de programación externos en el reporte	7	9
Capacidad para proveer familiaridad con Microsoft Office	8	9.5
Promedio	7.5	9.25
<b>Seguridad de la solución de reportes</b>		
Seguridad de la aplicación	9	9
Definición de reportes estratificados	9	9
Capacidad para enlazar la seguridad de la aplicación a la de acceso a datos de reportes	8	9
Seguridad de la solución de reportes	9	9
Promedio	8.75	9
<b>Integración y manejo del visor en la solución de reportes</b>		
Correspondencia del diseño con el resultado final	9	9
Integración con la aplicación del negocio	9	9
Capacidad para mantener características de presentación en Web	9	9.5
Soporte en diferentes formatos	9	9
Promedio	9	9.13
<b>Flexibilidad para establecer la conectividad de la solución de reportes</b>		

Soporte a diferentes motores de base de datos	10	9.5
Capacidad para compartir fuentes de datos entre reportes	8	9.5
Capacidad de incluir múltiples fuentes de datos en un mismo reporte	6	9.5
Elaboración de consultas dinámicas a partir de parámetros	6	9
Capacidad para incluir JavaBeans	9	0
<b>Promedio</b>	<b>7.8</b>	<b>7.5</b>
<b>Construcción de reportes</b>		
Disponibilidad de asistentes para simplificar la creación de reportes	9.5	9
Capacidad para incluir el diseñador de informes como parte de la aplicación	6.5	0
Capacidad para hacer uso de asistentes de forma integrada en Visual Studio .Net y Office	9	7
<b>Promedio</b>	<b>8.33</b>	<b>5.33</b>
<b>Programación de reportes</b>		
Definición de reportes programáticamente	8	9.5
<b>Promedio</b>	<b>8</b>	<b>9.5</b>
<b>Capacidad para definir reportes variados en una misma solución</b>		
Capacidad para definir reportes variados en una misma solución	8.5	9
<b>Promedio</b>	<b>8.5</b>	<b>9</b>
<b>Flexibilidad para establecer ordenamiento y agrupación sofisticados</b>		
Flexibilidad para establecer ordenamiento y agrupación sofisticados	8	9.5
<b>Promedio</b>	<b>8</b>	<b>9.5</b>
<b>Definición del formato del reporte</b>		
Definición del formato del reporte	9.5	9.3
<b>Promedio</b>	<b>9.5</b>	<b>9.3</b>
<b>Variedad y flexibilidad en la manipulación de objetos del informe</b>		
Capacidad para incluir objetos variados acorde a la naturaleza del reporte	9.5	9
Facilidad para definir fórmulas y cálculos sobre campos de la base	9	9
Flexibilidad en la ubicación de elementos en una matriz, totales, y funciones agregadas	8.5	9
Manejo de objetos compartidos en el proyecto para disminuir el peso	8	9.5
Capacidad para incluir enlaces desde los reportes	8	9
Capacidad para incluir textos con formato provenientes de la base de datos	9	7
<b>Promedio</b>	<b>8.67</b>	<b>8.75</b>
<b>Administración y programación referente al paso de parámetros</b>		
Capacidad para definir parámetros programáticamente	8	9.5
Capacidad para incluir valores disponibles de un parámetro proveniente de bd o de otras fuentes	6	9.5
Capacidad de incluir dependencia entre parámetros	9	9.5
<b>Promedio</b>	<b>7.67</b>	<b>9.50</b>
<b>Capacidad para incluir información de la BD</b>		
Capacidad para presentar toda la información de la bpd	9	9
Disponibilidad de sinónimos de una base de datos	9	8
<b>Promedio</b>	<b>9</b>	<b>8.5</b>
<b>Facilidad de navegación entre objetos y repositorios</b>		
Facilidad de navegación entre objetos y repositorios	9.5	8.5
<b>Promedio</b>	<b>9.5</b>	<b>8.5</b>
<b>Capacidad para incorporar características propias de la aplicación y cliente en la solución de reportes.</b>		
Capacidad de personalizar reportes y ambientes	9	9.5
<b>Promedio</b>	<b>9</b>	<b>9.5</b>

Los dos cuadros presentados, constituyen una cuantificación de las características y capacidades que tanto SAP Business Objects, como SQL Server Reporting Services ofrecen. Ambos productos son muy similares al crear y formatear informes. Crystal Reports, presenta ventajas para acceder a los datos de los mapas en los informes. Sin embargo, cuando trata de una flexibilidad general en los informes, SQL Server Reporting Services tiene la ventaja, ya que es totalmente ampliable mediante el uso de .NET.

En comparación con la edición para desarrolladores de Crystal Reports, SQL Server Reporting Services ofrece mucho más funcionalidad en lo que respecta a publicación, seguridad y distribución de informes. La decisión para optar por una u otra plataforma depende entonces de la presencia de SQL Server en el entorno, o si va usar otros lenguajes de desarrollo, tales como Java.

En los entornos cuyo motor de base de datos es SQL Server, es mucho más económico utilizar Reporting Services, especialmente si se está ejecutando desde el mismo servidor que la licencia de SQL Server. Con su extensibilidad y su capacidad de extraer y formatear datos de múltiples fuentes, SQL Server Reporting Services proporciona una excelente plataforma para la creación y entrega de informes a los usuarios de las aplicaciones.

### **2.13. La Tecnología .NET**

Para cumplir con uno de los principales objetivos que es la elaboración de una herramienta capaz de generar reportes ad hoc de manera fácil e intuitiva, es necesario seleccionar el entorno de programación a utilizar. De un breve análisis sobre algunas de las tecnologías como PHP, .NET, y Ruby. Se ha seleccionado la plataforma .NET, con su reciente lanzamiento Visual Studio 2008 y todas las potencialidades que abarca, las mismas que van de la mano con los objetivos del presente trabajo que es incorporar dinamismo a las funcionalidades y herramientas de programación.

Este año precisamente, se hizo oficial el lanzamiento de Visual Studio .Net, con el Framework 3.5, que a diferencia de la versión 3.0, si establece hitos en el manejo funcional de la programación.

Algunas de las características que es necesario tener presente sobre la tecnología .NET son<sup>1</sup>:

### **2.13.1. ASP.NET 3.5**

Cuando Microsoft lanzó ASP.NET 1.0, no pudo anticipar con cuanto entusiasmo sería recibida esta tecnología. ASP.NET rápidamente llegó a ser el estándar para el desarrollo de aplicaciones web con tecnologías Microsoft y sin un competidor lo suficientemente fuerte en plataformas de desarrollo web<sup>2</sup>.

Desde entonces, ASP.NET tuvo un menor lanzamiento (ASP.NET 1.1) y dos lanzamientos más significativos (ASP.NET 2.0 y ASP.NET 3.5).

### **ASP.NET 2.0**

ASP.NET 2.0 mantuvo la misma base por completo y se concentró en añadir características nuevas y de alto nivel como:

- Más controles enriquecidos: ASP.NET 2.0 introduce más de 40 nuevos controles, desde los más aguardados como un TreeView colapsable hasta un poderoso menú JavaScript.
- Master pages: Las páginas maestras son reutilizables como plantillas de páginas. Por ejemplo, se puede utilizar un master page para asegurarse que cada página web de la aplicación tenga la misma cabecera, pie y controles de navegación.
- Themes: Los temas permiten definir un conjunto estandarizado de características de apariencia para los controles web. Una vez definidos, se puede aplicar esos formatos a través de todo el sitio web para que lograr un look consistente.
- Seguridad y membership: ASP.NET 2.0 añade características relacionadas con seguridad, incluyendo soporte automático para almacenamiento de credenciales de

---

<sup>1</sup>Los contenidos siguientes están basados en (Troelsen, 2007), se hace referencia a figuras referentes a Framework 3.5 y Visual Studio 2008.

<sup>2</sup> "Estas estadísticas adoptadas son algo conflictivas, Netcraft (<http://www.netcraft.com>), reconocida analista de internet, afirma que el uso de ASP.NET corre ahora en más servidores que JSP. Este no es un dato amplio considerando el tamaño relativo de los sitios web, pero ASP.NET es la base de websites para un número significativo de más de 1000 empresas de alto tráfico como MySpace" (Troelsen, 2007).



usuario, y pre construye controles de seguridad para tareas comunes como inicio de sesión, registro, y recuperación de claves olvidadas.

- **Data Source Controls:** El modelo de data source permite definir como interactúa la página con una fuente de datos de forma declarada en el markup, en vez de tener que escribir el código equivalente de acceso a datos a mano. Se puede poblar un componente de datos customizado tan fácil se lo hace de forma directa desde la base de datos.
- **Partes Web:** Un tipo común de una aplicación web es el portal, el cual centraliza diferente información usando paneles separados en una página web. Las partes web proveen un framework completo de un portal pre construido con un layout flow-based, vistas configurables, e incluso soporte drag-and-drop.
- **Perfiles:** Esta característica permite almacenar información específica de usuario en una base de datos sin escribir ningún código de base de datos.

### **ASP.NET 3.5**

Microsoft utilizó el nombre .NET Framework 3.0 para lanzar nuevas tecnologías como WPF (Windows Presentation Foundation), una nueva interfaz de usuario para construcción de cliente enriquecidos, WCF (Windows Communication Foundation), una tecnología para construcción de servicios orientados en mensajes, y WF (Windows Workflow Foundation), una tecnología que permite modelar un proceso de negocios complejo como una serie de acciones (opcionalmente usando un visual flowchart como diseñador). Sin embargo el Framework .NET 3.0 no incluye una nueva versión de CLR o ASP.NET. En su lugar, el próximo lanzamiento de ASP.NET fue incluido en el Framework .NET 3.5.

Comparado con ASP.NET 2.0, ASP.NET 3.5 es una evolución más gradual. Sus nuevas características están concentradas en dos áreas: LINQ y Ajax.

### 2.13.2. LINQ

LINQ (Language Integrated Query) es un conjunto de extensiones para los lenguajes C# y Visual Basic. Permite escribir código C# o Visual Basic que manipula en memoria datos de la misma forma que lo haría en un query a una base de datos.

Técnicamente, LINQ define cerca de 40 operadores query, tales como select, from, in, where, y orderby (en C#), dichos operadores permiten programar consultas. Sin embargo, hay varios tipos de datos sobre los cuales estas consultas pueden ser ejecutadas, y cada tipo de dato requiere un toque propio de LINQ.

La rama fundamental de LINQ es LINQ to Objects, que permite tomar una colección de objetos y ejecutar un query que extrae alguno de los detalles desde algunos de los objetos. LINQ to Objects no es específicamente ASP.NET. En otras palabras, se puede utilizar en una página web de la misma manera que se puede utilizar en cualquier otro tipo de aplicación .NET.

Junto con LINQ to Objects está LINQ to DataSet, el cual provee similares características para consultas en memoria de un objeto dataset, y LINQ to XML, el cual trabaja con datos XML. No obstante la rama de LINQ que atrajo mayor atención es LINQ to SQL, que permite utilizar la sintaxis LINQ para ejecutar una consulta sobre un servidor de base de datos SQL. Esencialmente, LINQ to SQL crea una consulta apropiadamente parametrizada basada en el código y ejecuta la consulta cuando se intenta acceder a los resultados del query. No se necesita escribir ningún código de acceso a datos o utilizar los tradicionales objetos ADO.NET.

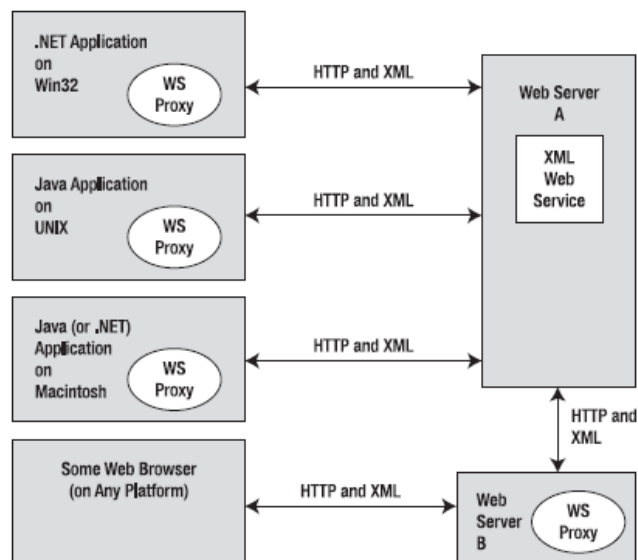
LINQ to Objects, LINQ to DataSet, y LINQ to XML son características que complementan ASP.NET, y no están sujetas de ninguna manera a él. Sin embargo ASP.NET incluye mejoras para el soporte de LINQ to SQL, incluyendo un control de fuente de datos que deja ejecutar una consulta a través de LINQ to SQL y poblar los resultados en un web control, sin requerir código extra.

## 2.14. Servicios Web y Windows Communication Foundation

Los sistemas operativos Windows, históricamente han proporcionado un número de APIs para construir sistemas distribuidos.

Los servicios Web marcaron sin duda, uno de los avances más importantes en el desarrollo de aplicaciones distribuidas, detalles como el empleo de estándares HTTP y SOAP los hicieron capaces de trabajar incluso con firewalls y software NAT.

Por supuesto no es un API perfecto, una de las debilidades de los servicios web es el hecho que pueden sufrir algún deterioro de performance y por ende no son la solución ideal para aplicaciones locales donde el protocolo TCP y formato binario de datos pueden ser usados sin restricción.



**Figura VIII.** Interoperabilidad de los servicios web XML.

Si se está construyendo un sistema distribuido constituido por un conjunto de aplicaciones corriendo en la misma máquina física, se puede hacer uso de los denominados Pipes, Sockets, and peer-to-peer P2P services, los cuales aceleran el intercambio de datos entre aplicaciones en la misma máquina.

Dado que cada API tiene su propio modelo de programación, y conjunto de utilitarios para configuración entre otras cosas, si se decide hacer una migración de un modelo .NET

Remoting a Web Services por ejemplo, se tendría que hacer una reingeniería del código base.

Windows Communication Foundation es un toolkit introducido con .NET 3.0 que integra las anteriores tecnologías distribuidas dentro de un API simplificado primariamente por medio del namespace System.ServiceModel. Utilizando WCF, se puede publicar los servicios a consumidores utilizando una variedad de técnica. Por ejemplo si se está construyendo una aplicación local donde todas las máquinas están sobre Windows, se puede utilizar varios protocolos TCP para asegurar el mejor rendimiento posible. Este mismo servicio puede también ser publicado utilizando un servicio Web XML para permitir a consumidores externos asegurar su funcionalidad a pesar del lenguaje de programación o el sistema operativo.

#### **2.14.1. Características de WCF**

La interoperabilidad e integración de diversos APIs, son dos aspectos importantes de WCF, adicionalmente esta tecnología proporciona una fábrica de software enriquecida que complementa las remoting technologies.

- Soporte para strongly typed también conocidos como mensajes sin tipo. Esto permite a las aplicaciones .NET compartir tipos personalizados eficientemente.
- Soporte para las últimas especificaciones de web service (WS-\*).
- Un modelo completamente integrado tanto con protocolos de seguridad nativos Win32/.NET y numerosas técnicas neutrales de seguridad construidas sobre estándares de web services.
- Soporte para manejo de sesión, como técnicas para el manejo de estados, así como soporte para mensajes one-way sin estado.

WFC también proporciona facilidades para captura y log de eventos, contadores de rendimiento, un modelo de publicación y suscripción de eventos, y soporte transaccional.

### 2.14.2. Arquitectura Orientada a Servicios

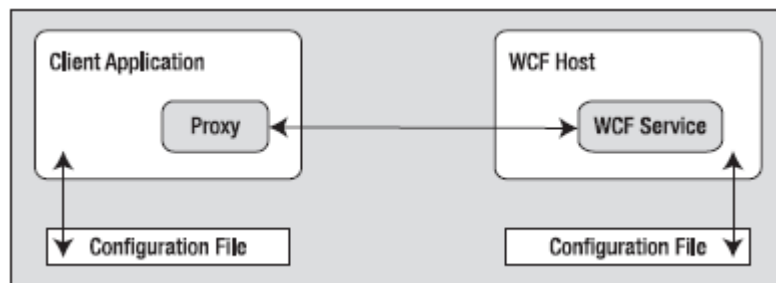
Otro de los beneficios de WCF es que está basado en los principios de diseño establecidos por la arquitectura orientada a servicios (SOA). SOA es una forma de diseñar un sistema distribuido donde varios servicios autónomos trabajan en conjunto por medio del paso de mensajes direccionados (tanto en máquinas en red como dos procesos en una misma máquina) usando interfaces bien definidas.

Cuando WCF fue diseñado, el equipo de WCF lo hizo en función de los principios de diseño de SOA, Principio 1: El direccionamiento es explícito, Principio 2: Los servicios son autónomos. Principio 3: Los servicios se comunican vía contrato, no implementación. Principio 4: La compatibilidad de Servicios está basada en políticas.

### 2.14.3. Composición básica de una aplicación WCF

Cuando se construye un sistema distribuido WCF, se tiene que crear tres assembles interrelacionados.

- WCF Service assemble: Este \*.dl, contiene las clases e interfaces que representan la funcionalidad completa que se está publicando a consumidores externos.
- WCF Service host: Este modulo software es una entidad que presenta el assembly del servicio WFC.
- WCF client: Esta es la aplicación que accede a la funcionalidad del servicio a través de un proxy intervensor.



**Figura IX.** Relación entre las tres capas de WCF en alto nivel

## **2.15. Serialización y deserialización de objetos**

Utilizando la serialización de objetos se puede persistir o recuperar el estado de un objeto desde o hasta cualquier tipo `System.IO.Streamderived` (incluyendo el tipo `IsolatedStorageFileStream` type).

La capacidad para serializar tipos es crítica cuando se intenta copiar un objeto a una máquina remota vía varias tecnologías remotas tales como la paca .NET remoting, XML web services, y Windows Communication Foundation. Entendiendo desde luego que la serialización es bastante útil en otros ámbitos y juega un papel muy importante en las aplicaciones .NET, sean distribuidas o no.

El término serialización describe el proceso de mantener (y posiblemente transferir) el estado de un objeto en una dirección de memoria o un archivo. La secuencia de datos capturada contiene toda la información necesaria para reconstruir (o deserializar) el estado del objeto para usarlo luego. Usando esta tecnología, es trivial almacenar grandes cantidades de datos. En muchos casos, almacenar los datos de la aplicación utilizando servicios de serialización resulta un menor esfuerzo que hacer uso de lecturas y escrituras encontradas en el namespace `System.IO`.

Mientras esto es ciertamente más viable, si se guarda su propia información haciendo la clase `UserPrefs` serializable con el atributo `[Serializable]`.

Mientras es mucho más simple mantener objetos utilizando la serialización de objetos .NET, los procesos utilizados por debajo son bastantes sofisticados. Por ejemplo, cuando un objeto es capturado en un determinado instante, los datos asociados (datos de clases base, objetos contenidos, etc.) también son serializados automáticamente. Un conjunto de objetos interrelacionados es representado utilizando un `object graph`.

Los servicios de serialización .NET también permiten mantener un `object graph` en una variedad de formatos. También se puede almacenar un `object graph` en formatos SOAP o XML utilizando otros tipos. Estos formatos pueden ser bastante útiles cuando se desea asegurar que los objetos almacenados viajen bien a través de sistemas operativos, lenguajes y arquitecturas.

### 2.15.1. Selección del Formato de Serialización

Una vez que se tenga configurado los tipos que participarán en el esquema de serialización mediante la aplicación de los atributos adecuados, el siguiente paso es determinar cual formato (binary, SOAP, o XML) deberá ser usado cuando se almacene el estado del objeto. Cada posibilidad está representada en las siguientes clases:

- **BinaryFormatter**. Serializa el estado de los objetos a una cadena utilizando un formato binario compacto. Este tipo está definido en el namespace `System.Runtime.Serialization.Formatters.Binary` que es parte de `mscorlib.dll`.
- **SoapFormatter**. Almacena el estado de un objeto como un mensaje SOAP. Este tipo está definido dentro del namespace `System.Runtime.Serialization.Formatters.Soap` namespace, que está dentro de un assembly separado. Por tanto, para formatear un object graph dentro de un mensaje SOAP, primero se debe colocar una referencia a `System.Runtime.Serialization.Formatters.Soap.dll`.
- **XmlSerializer**. Si se desea almacenar un árbol de objetos como un documento XML, se tiene el tipo `XmlSerializer`. Para usar este tipo solo se necesita especificar que se utilizará el namespace `System.Xml.Serialization` y un conjunto de referencias al assembly `System.Xml.dll`

### 2.15.2. Serialización usando BinaryFormatter

Los métodos del tipo `BinaryFormatter` son:

- `Serialize()`: Almacena el objeto a un stream específico como una secuencia de bytes.
- `Deserialize()`: Convierte una secuencia de bytes almacenada en un object graph.

## 2.16. Microsoft Visual Studio 2008

La entrega de Microsoft, Visual Studio 2008, se basa en el precepto de entregar a las organizaciones la capacidad para encontrar la manera más fácil para capturar y analizar la información para ayudarlos a la toma de decisiones. Visual Studio 2008 habilita a las

organizaciones de todo tamaño crear aplicaciones más seguras y administrables tomando las ventajas de Windows Vista™ y el Office System 2007.

Se basa en tres pilares fundamentales:

- **Desarrollo rápido de aplicaciones.** Productividad mediante el uso de AJAX, se tiene la ventaja de tecnología IntelliSense en variables, objetos y métodos Visual Studio 2008. Pueden depurar código JavaScript colocando breakpoints dentro de Visual Studio— una capacidad que trabaja con los browsers Web más populares.
- **Colaboración efectiva de equipo.** Los diseñadores de interfaz y desarrolladores software pueden trabajar juntos es una ambiente sin roces, con proyectos compartidos, código y diseños. Los diseñadores pueden utilizar Microsoft Expresión Web para diseñar una interfaz de usuario y entonces girarlo a los desarrolladores con la seguridad que el diseño UI y la lógica de negocio desarrollada subsecuentemente se mantengan intactos.
- **Incursionar a través de la experiencia de los usuarios.** El soporte mejorado para AJAX posibilita a los desarrolladores entregar aplicaciones web enriquecidas y con mejor respuesta, eliminando la necesidad de recargar una página completa cuando el usuario realiza un cambio.

## 2.17. Microsoft C# 2008

C# 2008 proporciona un número de características muy interesantes que están relacionadas con los lenguajes funcionales.

En ésta versión se han añadido varios conceptos como variables locales de tipo implícito, que facilitan la interacción con la metodología LINQ, propiedades automáticas, métodos parciales, métodos extendidos (que permiten añadir nueva funcionalidad a tipos ya compilados), tipos anónimos , “shapes” que son bastante útiles al momento de modelar un tipo para un uso limitado en un programa, y también se hace énfasis en uso de delegados, tipos genéricos, sintaxis de inicialización de objetos, entre otros aspectos que han hecho atractivo el uso y migración de proyectos hacia esta versión.



### **2.17.1. Características del lenguaje C# 2008**

Desde el release de .NET 2.0, el lenguaje de programación C# ofrece soporte para genéricos. La programación genérica mejora el rendimiento de la aplicación y seguridad de tipos. No solo que se cuenta con el namespace `System.Collections.Generic`, sino que también se puede construir métodos y tipos genéricos propios (con algunas restricciones)

Esta versión impulsa también la explotación del uso de delegados. Un delegado es solo un objeto que apunta a otros métodos en la aplicación. Usando este patrón se puede construir sistemas que permitan que múltiples objetos se acoplen en una conversación de dos vías. También aparece el concepto C# `event keyword`, el cual es usado para simplificar la manipulación de programación con delegados. Se ha hecho hincapié en la programación funcional, introduciendo el operador lambda, explorando la conexión entre delegados, métodos anónimos y expresiones lambda.

En general se introduce técnicas avanzadas de programación, como sobrecarga de operadores y rutinas de conversión personalizadas (tanto implícitas como explícitas) entre tipos.

### **2.17.2. Variables locales de tipos implícitos.**

Tradicionalmente la declaración de variables se ha realizado de una forma explícita, especificando el tipo de dato, el nombre de la variable y el valor inicial.

C# 2008, introduce una nueva palabra clave, `var`, la cuál es usada en lugar de especificar un tipo de dato formal (como `int`, `bool` o `string`). Cuando se la usa, el compilador automáticamente infiere el tipo de dato basado en el valor inicial que se le dé a la variable, por ejemplo: `var myInt = 0; var myBool = true; var myString = "Time, marches on...";` En estos casos, el compilador infiere los tipos de datos `myInt` como un `System.Int32`, `myBool` como un `System.Boolean`, y `myString` es identificado como `System.String`, dado el valor inicial asignado.

Esta declaración implícita se puede realizar en cualquier tipo de dato incluyendo arrays, tipos genéricos, y tipos personalizados.

Las variables con tipos implícitos son bastante útiles sin embargo no es recomendable abusar de su uso, debido a que un excesivo manejo de las mismas provocarían confusión a la hora de entender un código. Sin embargo, existen otras aplicaciones en donde utilizar la palabra clave `var`, facilita increíblemente el desarrollo de lógica funcional, como en el caso de las expresiones lambda o LINQ.

### 2.17.3. Propiedades Automáticas

En los lenguajes de programación de .NET, se prefiere el uso de propiedades para asegurar la obtención y asignar campos privados de un tipo, utilizando los métodos tradicionales `GetXX()` y `SetXX()`.

La definición de una propiedad en C# normalmente obedece a la siguiente sintaxis:

```
private string nombreField = string.Empty;
public string NombreProperty
{
    get{return nombreField;}
    set{ nombreField = value;}
}
```

Esta sintaxis a pesar de ser del todo simple, todavía sigue siendo abrumante si es que se requiere la definición de múltiples campos considerando que el objetivo de la sintaxis manejada es proveer una encapsulación simple de los campos.

Para simplificar el proceso de proporcionar encapsulación simple de datos, C # 2008 ahora proporciona el denominado “automatic property syntax” (sintaxis de propiedades automática). Como su nombre lo indica, esta característica permite aligerar el trabajo de definir campos privados y la propiedad respectada al compilador utilizando una nueva forma de sintaxis.

La sintaxis para el ejemplo propuesto es la siguiente:

```
public string NombreProperty { get; set; }
```

En el modelo tradicional de encapsulación se pueden hacer uso de un campo privado de forma directa dentro de la clase que lo ha definido, esto no es posible con el uso de propiedades automáticas.

#### **2.17.4. Métodos de Extensión**

Como es sabido, una vez que un tipo es definido y compilado dentro de un assembly .NET, esta definición, en muchos de los casos, es final. La única manera de añadir nuevos miembros, o actualizar o eliminar miembros existentes es reescribir y recompilar el código base en un assembly actualizado (o tomando medidas mucho más complejas como el uso del namespace System.Reflection.Emit para rediseñar dinámicamente un tipo compilado en memoria).

C# 2008, ahora hace posible definir métodos de extensión. En pocas palabras los métodos de extensión permiten que tipos compilados existentes (clases, estructuras, o implementación de interfaces), así como tipos que están siendo compilados (como tipos en un proyecto que tienen métodos de extensión), adquieran nueva funcionalidad sin necesidad de actualizar directamente el tipo a ser extendido.

Esta técnica puede resultar bastante útil cuando se necesita inyectar nueva funcionalidad a tipos de los cuales no se tiene el código fuente. Así como también en los casos en que se necesite forzar a un tipo para que soporte un conjunto de miembros (principio fundamental del polimorfismo).

Es importante puntualizar que los métodos de extensión no modifican literalmente el código base. Esta técnica solo añade miembros a un tipo dentro del contexto de la aplicación actual.

Cuando se define métodos de extensión, la primera restricción es que estos deben definirse dentro de una clase estática, consecuentemente cada método de extensión debe también ser declarado con el keyword static.

#### **El alcance de un método de extensión**

Es importante puntualizar que los métodos de extensión a diferencia de un método "normal", no tienen acceso directo a los miembros del tipo que están extendiendo, dicho de otro modo la extensión no es herencia.

Aunque pueda parecer que los métodos de extensión son globales por naturaleza, están limitados a los namespaces que los definan o a los namespaces que los importe.

### **El IntelliSense de los Métodos de Extensión**

Dado que se puede crear docenas de métodos de extensión para un mismo tipo, para evitar el riesgo de confusión al distinguir los métodos de extensión de los originales, el mecanismo de intellisense de Visual Studio 2008, muestra un ícono único delante de los métodos de extensión. Es esta forma existe un método amigable de recordar que métodos fueron definidos fuera de la definición original de la clase.

#### **2.17.5. Métodos Parciales**

Desde el release de .NET 2.0, se puede construir definiciones parciales de clases utilizando el keyword `partial`. Recordemos que esta pequeña sintaxis permite particionar la implementación completa de un tipo a través de múltiples archivos de código (u otras ubicaciones como memoria). Siempre y cuando cada aspecto del tipo parcial tenga el mismo nombre calificador completo, el resultado final es un tipo clase "normal" compilado en el assembly construido.

C#2008 amplía el alcance del keyword `partial`, en que ahora puede ser aplicado a un nivel de métodos. En resumidas cuentas, esto permite prototipar un método en un archivo, pero implementarlo en otro. Si se tiene bases de C++, este puede recordar el concepto de un archivo header/implementation de C++. Sin embargo, los métodos parciales de C# tienen un importante número de restricciones.

- Los métodos parciales solo pueden ser definidos dentro de clases parciales.
- Los métodos parciales deben retornar `void`.
- Los métodos parciales pueden ser estáticos o a nivel de instancia.
- Los métodos parciales pueden tener argumentos (incluyendo a los parámetros modificados por los calificadores `this`, `ref` o `params`, pero no el modificador `out`).
- Los métodos parciales son siempre privados de manera implícita.

Los métodos parciales pueden o no estar contenidos en el assembly que se genere.

### 2.17.6. Sintaxis de Inicialización de Objetos

C#2008 ofrece una nueva forma de llenar el estado de una clase o estructura nueva, denominada object initializer syntax o sintaxis de inicialización de objetos. Usando ésta técnica, es posible crear una nueva variable y asignarle un conjunto de propiedades y/o campos públicos en pocas líneas de código. Sintácticamente, una inicialización de objetos consiste de una lista de valores especificados separados por coma, entre llaves { y }. Cada miembro de la lista de inicialización se mapea con el nombre del campo público o propiedad de el objeto a ser inicializado. Esta sintaxis puede verse en el siguiente ejemplo:

```
var Point = new Point { X = 30, Y = 30 };
```

### 2.17.7. Inicialización de Colecciones

Cercano al concepto de la sintaxis de inicialización de objetos esta la inicialización de colecciones. Esta sintaxis hace posible poblar un contenedor (como un ArrayList o List<T>) con ítems utilizando un simple Array. Por ejemplo:

**//Inicialización de un array de objetos.**

```
int[] myArrayOfInts = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

**// Inicialización de una lista generic de enteros.**

```
List<int> myGenericList = new List<int> { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

**// Inicialización de un ArrayList con datos numéricos.**

```
ArrayList myList = new ArrayList { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

### 2.17.8. Tipos Anónimos

En la programación orientada a objetos, se conoce los beneficios de la definición de clases para representar el estado y funcionalidad de una entidad de programación dada. Sin duda, cuando se necesita definir una clase que se pretende sea reutilizada a través de proyectos y proporcione numerosas porciones de funcionalidad a través de un conjunto de métodos, eventos, propiedades y constructores personalizados, la creación de una nueva clase C# es una práctica a menudo obligatoria.

No obstante, hay otros momentos en la programación cuando se necesita definir una clase simplemente para modelar un set de puntos de datos encapsulados sin ningún método asociado, eventos o funcionalidad personalizada. Más aún, si el tipo es usado solo internamente a la aplicación actual y no se pretende reutilizarlo. Si se necesita un tipo "temporal", en las anteriores versiones de C#, se requiere a pesar de todo construir una nueva definición de una clase.

Mientras la construcción de una clase no es tan compleja, puede ser una labor intensiva si se está intentando encapsular más que un conjunto de miembros (característica solventada mediante las propiedades automáticas). En C# 2008, se dispone de un atajo para esta problemática denominada tipos anónimos (anonymous types), que en cierta forma es una extensión natural de la sintaxis de miembros anónimos.

Cuando se define un tipo anónimo, se lo hace por medio del uso del nuevo keyword var en conjunto con la sintaxis de inicialización de objetos. Para ilustrarlo consideremos el siguiente ejemplo:

```
static void Main(string[] args)
{
    // Genera un tipo anónimo representando un auto.
    var myAuto = new { Color = "Black", Marca = "Ford", VelocidadActual = 55 };
    // Ahora muestra el color y la marca.
    Console.WriteLine("Mi auto es un {0} {1}.", myCar.Color, myCar.Marca);
}
```

Nótese que la variable myAuto debe ser implícitamente un tipo, el cual compensa el que no se haya modelado el concepto de un automóvil usando la definición de clases. En tiempo de compilación, el compilador autogenerara una clase con un nombre único en nuestro beneficio. Dado que el nombre de la clase no es visible desde C#, el uso de tipos implícitos usando el keyword var es obligatorio. También hay que considerar que se debe especificar (utilizando la sintaxis de inicialización de objetos) el conjunto de propiedad que modelan los datos que se está intentando encapsular. Una vez definidos, esos valores pueden ser obtenidos usando la sintaxis de invocación de propiedades standard de C#.

Todos los tipos anónimos son derivados automáticamente de System.Object, y por tanto soportan cada uno de los miembros provistos por esta clase base. Dado esto, se puede

invocar a los métodos ToString(), GetHashCode(), Equals(), or GetType en el objeto de tipo implícito.

### **La semántica de igualdad de tipos anónimos**

Para determinar la igualdad de tipos anónimos se debe utilizar el método Equals, el mismo que realiza una comparación semántica basada en valor (*value-based semantics*) validando cada campo en el tipo anónimo.

Si se utiliza el operador de comparación de C#, al comparar dos tipos anónimos con la misma definición sin embargo, no se obtendrá un resultado de igualdad favorable debido a que los tipos anónimos no reciben una sobrecarga de igualdad de los operadores (== y !=) de C#. Dado esto cuando se consulte por la igualdad de tipos anónimos utilizando los operadores de igualdad de C#, las referencias, no los valores de los objetos, son los que serán comparados.

Algo importante de los tipos anónimos es que cuando se tiene dos definiciones de las mismas características el compilador solo generará una misma clase.

El uso de los tipos anónimos debe ser moderado, si es posible, únicamente cuando se haga uso de la tecnología LINQ. Es recomendable no abandonar el uso de clases y estructuras definidas claramente, debido a que los tipos anónimos tienen numerosas debilidades entre las que se incluyen las siguientes:

- No se controla el nombre del tipo anónimo.
- Los tipos anónimos siempre derivan de System.Object.
- Los campos y propiedades de un tipo anónimo siempre son solo de lectura.
- Los tipos anónimos no soportan eventos, métodos u operadores personalizados
- Los tipos anónimos siempre son sellados de forma implícita.
- Los tipos anónimos son creados siempre utilizando el constructor por default.

A pesar de estas limitaciones, cuando se programa con la tecnología LINQ, se encuentra muchas aplicaciones sumamente útiles de los tipos anónimos.

## 2.18. LINQ APIs

Con LINQ se puede hacer mucho más que consultas SQL embebidas, también está lejos de ser otro conjunto de reglas para modelado de datos. LINQ es una metodología que simplifica y unifica la implementación de cualquier clase de acceso a datos. LINQ no fuerza el uso de una arquitectura en específico, más bien facilita la implementación de las arquitecturas existentes para acceso a datos.

Como toda herramienta puede ser utilizada correcta o incorrectamente. Para obtener lo mejor de LINQ se debe tener suficiente conocimiento del mismo.

Hoy en día la administración de datos en un programa puede provenir de una larga lista de diferentes dominios de datos: un array, un object graph, un documento XML, una base de datos, un archivo de texto, una clave del registro, mensajes e-mail, mensajes SOAP, archivos Excel, entre otros. Cada dominio de datos tiene su modelo de acceso propio. Cuando se tiene una consulta a una base de datos, típicamente se utiliza SQL. Se navega en datos XML con el modelo DOM o XQuery.

Se itera un array y construye algoritmos para navegar un object graph. Se utiliza APIs específicas para acceder a otros dominios de datos como archivos Excel, mensajes de correo, o el registro de Windows. En conclusión se tiene distintos modelos de programación para acceder a diferentes fuentes de datos.

La unificación de estas técnicas de acceso a datos en un solo modelo completo se ha intentado de diversas formas. Sin embargo aún se debe utilizar un lenguaje como SQL para acceder a los datos representados a través de un modelo relacional.

En ocasiones los datos son representados de forma más efectiva mediante un modelo jerárquico o gráfico que mediante un modelo relacional. Más aún, si el modelo de datos no está ligado al lenguaje, probablemente se tenga que manejar diferentes tipos de sistemas. Todas esas diferencias crean un muro entre los datos y el código. LINQ intenta solventar estos problemas, ofreciendo una manera uniforme para acceder y administrar los datos sin forzar la adopción de un modelo "one size fits all" (igual para todos). LINQ hace uso de las similitudes



entre las operaciones entre esos modelos de datos en lugar de emparejar las diferentes estructuras entre ellos.

### 2.18.1. LINQ

LINQ es un modelo de programación que introduce consultas como un concepto de primera clase dentro de cualquier lenguaje .NET. Sin embargo, el soporte completo para LINQ requiere algunas extensiones en el uso del lenguaje. Dichas extensiones aumentan la productividad, consecuente proporciona una sintaxis expresiva, reducida, y significativa para manipular los datos<sup>1</sup>.

El siguiente ejemplo es una consulta LINQ simple, para una típica solución que retorna los nombres de clientes en Perú.

```
var query =  
from c in Customers  
where c.Country == "Peru"  
select c.CompanyName;
```

El resultado de esta consulta es una lista de strings. Se puede enumerar estos valores con un ciclo foreach en C#.

```
foreach ( string name in query ) {  
Console.WriteLine( name );  
}
```

Tanto la definición de la consulta, y el ciclo foreach son sentencias regulares de C#. En este punto se podría pensar que esta LINQ consulta es una nueva forma de SQL embebido. En lo absoluto, la misma consulta y el consecuente ciclo foreach, puede ser aplicada a una base SQL, a un DataSet, a un array de objetos en memoria, o muchas clases de datos. También puede ser una instancia de una clase que describe una tabla física en una base de datos relacional. O puede ser una instancia de una clase que describe el modelo conceptual y que esta mapeado a una base de datos relacional.

La sintaxis como SQL usada en LINQ se denomina query expression. Los lenguajes que implementan SQL embebido definen solo una sintaxis simplificada para poner sentencias SQL dentro de un lenguaje distinto, pero esas sentencias no están integradas con la

---

<sup>1</sup> Los conceptos y ejemplos toman como base al libro *Introducing Microsoft LINQ* (Pialorsi & Russo, 2007)

sintaxis nativa del lenguaje y el sistema. Por ejemplo no se puede invocar a una función escrita en el lenguaje anfitrión en medio de una sentencia SQL, esto sí es posible en LINQ y lo mejor de todo es que no se limita a consultas en la base de datos, como lo hace el SQL embebido.

## 2.18.2. Funcionamiento de LINQ

Una vez comprendido el concepto de tener una sintaxis que integra consultas dentro de un lenguaje, se puede ver cómo trabaja en los siguientes ejemplos.

```
Customer[] Customers = GetCustomers();  
var query =  
from c in Customers  
where c.Country == "Peru"  
select c;
```

el compilador genera este código:

```
Customer[] Customers = GetCustomers();  
IEnumerable<Customer> query =  
Customers  
.Where( c => c.Country == "Peru" );
```

Si se realiza la consulta:

```
var query =  
from c in Customers  
where c.Country == "Italy"  
orderby c.Name  
select new { c.Name, c.City };
```

El código generado es:

```
var query =  
Customers  
.Where( c => c.Country == "Italy" );  
.OrderBy( c => c.Name )  
.Select( c => new { c.Name, c.City } );
```

Como se puede ver, el código, aparentemente llama a miembros instanciados en el objeto resultante de la llamada previa. Esta característica está regulada por las características de los métodos de extensión del lenguaje anfitrión, en este caso C#. La implementación de los métodos Where, OrderBy y Select, llamados en el ejemplo, dependen del tipo de Customers y los namespaces especificados en las sentencias using. Los métodos de extensión son una característica fundamental que es usada por LINQ para operar con diferentes dominios de datos utilizando la misma sintaxis.

Otro concepto importante es el cronometraje de operaciones sobre los datos. En general, una consulta LINQ no es ejecutada hasta que esta accede al resultado de datos, porque

describe un conjunto de operaciones que serán ejecutadas cuando sea necesario. El acceso al resultado de la consulta no es el trabajo real. Esto puede ilustrarse en el caso de un ciclo foreach.

Hay métodos que iteran un resultado de una consulta LINQ, produciendo una copia permanente de datos en memoria. Por ejemplo, el método ToList produce una colección tipo List<T>:

Cuando la consulta LINQ opera sobre datos que están en una base de datos relacional (como Microsoft SQL Server), la consulta LINQ genera un equivalente de sentencias SQL en lugar de operar con copias de tablas en memoria. La ejecución de la consulta sobre la base de datos se suspende hasta el primer acceso al resultado de la consulta. Por consiguiente, en los dos últimos ejemplos Customers pudo haber sido un tipo Table<Customer> (una tabla física en una base de datos relacional) o un tipo ObjectQuery<Customer> (una entidad conceptual mapeada a una base de datos relacional), la consulta equivalente SQL no será enviada a la base de datos hasta que el ciclo del foreach sea ejecutado o el método ToList sea invocado. La consulta LINQ puede ser manipulada y compuesta en diferentes modos entre tanto.

### **2.18.3. Modelo Relacional vs Modelo Jerárquico/Gráfico**

A primera vista, LINQ puede parecer otro dialecto SQL. Esta similitud tiene su raíz en la forma en que una consulta LINQ puede describir una relación entre entidades tal como un SQL join.

```
var query =  
from c in Customers  
join o in Orders  
on c.CustomerID equals o.CustomerID  
select new { c.CustomerID, c.CompanyName, o.OrderID };
```

Esto es similar a la manera regular de consultar datos en un modelo relacional. Sin embargo, LINQ no está limitado a un único dominio de datos a diferencia del modelo relacional. En un modelo jerárquico, suponiendo que cada customer tenga su propio conjunto de órdenes (orders) y cada orden tiene su propia lista de productos (products). En

LINQ se puede obtener la lista de productos ordenados por cada customer de la siguiente forma.

```
var query =  
from c in Customers  
from o in c.Orders  
select new { c.Name, o.Quantity, o.Product.ProductName };
```

La consulta anterior no contiene joins. La relación entre Customers y Orders está expresada en la segunda cláusula from, la cual usa c.Orders para decir “obtener todas las ordenes (Orders) de c Customer”. La relación entre Orders y Products está expresada por el miembro Product de la instancia Order. El resultado proyecta el nombre del producto para cada fila de orden utilizando o.Product.ProductName.

Las relaciones jerárquicas están expresadas en definiciones de tipos a través de referencias a otros objetos. Para soportar la consulta anterior, se debe tener clases similares a:

```
public class Customer {  
    public string Name;  
    public string City;  
    public Order[] Orders;  
}  
public struct Order {  
    public int Quantity;  
    public Product Product;  
}  
public class Product {  
    public int IdProduct;  
    public decimal Price;  
    public string ProductName;  
}
```

Disponibles en las bases de datos ejemplo de SQL Server.

Sin embargo, se puede tener cambios como el querer usar la misma instancia Product para diferentes Orders del mismo producto. Probablemente se quiera también filtrar Orders o Products, sin acceder a ellos a través de Customer. Un escenario común es el siguiente:

```
public class Customer {  
    public string Name;  
    public string City;  
    public Order[] Orders;  
}  
public struct Order {  
    public int Quantity;  
    public Product Product;  
    public Customer Customer;  
}  
public class Product {  
    public int IdProduct;  
    public decimal Price;  
    public string ProductName;  
    public Order[] Orders;  
}
```

Teniendo un array de productos declarado como sigue:

```
Product[] products;
```

Se puede consulta el diagrama de objetos, preguntando por la lista de ordenes de un único producto con un ID igual a 3.

```
var query =  
from p in products  
where p.IdProduct == 3  
from o in p.Orders  
select o;
```

Con el mismo lenguaje de consulta, se está consultando diferentes modelos de datos. Cuando no se tiene una relación definida entre las entidades utilizadas en la consulta, se puede contar con subconsultas y joins que están disponibles en la sintaxis LINQ así como en el lenguaje SQL. Sin embargo, cuando el modelo de datos ya define relaciones de entidades, se puede hacer uso de ellas, evitando replicación (con posibles errores) de la misma información en múltiples lugares.

Si se tiene relaciones de entidades en el modelo de datos, se puede utilizar también relaciones explícitas en una consulta LINQ, por ejemplo, cuando se quiere forzar alguna condición, o cuando simplemente se quiere relacionar entidades que no tienen relaciones nativas. Por ejemplo, si se quiere buscar customers y suppliers que vivan en la misma ciudad. El modelo de datos puede no puede proveer una relación explícita entre esos atributos, pero se puede escribir lo siguiente:

```
var query =  
from c in Customers  
join s in Suppliers  
on c.City equals s.City  
select new { c.City, c.Name, SupplierName = s.Name };  
El resultado:
```

```
City=Torino Name=Marco SupplierName=Trucker  
City=Dallas Name=James SupplierName=FastDelivery  
City=Dallas Name=James SupplierName=Horizon  
City=Seattle Name=Frank SupplierName=WayFaster
```

#### **2.18.4. Manejo de XML**

LINQ tiene un conjunto diferente de clases y extensiones para soportar el manejo de datos XML. Se puede crear algunos ejemplos utilizando el siguiente escenario. Usando las clases estándar Microsoft .NET 2.0 System.Xml, se puede cargar un archivo XML usando un enfoque DOM o se puede analizar su contenido utilizando una implementación XmlReader.

Independientemente de la solución que se escoja, se debe tomar en cuenta nodos y tipos de nodos, namespaces XML, y todos los elementos relacionados con el mundo XML.

A muchos desarrolladores no les agrada trabajar con XML porque ello requiere, el conocimiento de otras estructuras de dominios de datos y sintaxis de uso.

LINQ to XML facilita el tratamiento de documentos XML, utilizando un nuevo enfoque para el tratamiento de elementos y nodos, facilitando la carga y actualización de datos, y hace posible la utilización de las operaciones de consulta, filtrado, ordenamiento y búsqueda de LINQ sobre documentos XML completos. El resultado de estas consultas puede ser utilizado para cargar transparentemente una lista de entidades y luego se puede utilizar LINQ to SQL para enviar los cambios en una capa de base de datos física.

Esta solución mantiene la sintaxis XML sin perder la estabilidad del código escrito y transforma un conjunto de entidades seleccionadas vía LINQ to SQL en un XML InfoSet.

### **2.18.5. Integración de lenguajes**

La integración de lenguajes es un aspecto fundamental de LINQ. La parte más visible es la característica de expresión de consultas, la cual está presente en C# 3.0 y Visual Basic 9.0.

Que permite escribir código como el siguiente:

```
var query =  
from c in Customers  
where c.Country == "Italy"  
orderby c.Name  
select new { c.Name, c.City };
```

en lugar de escribir este código

```
var query =  
Customers  
.Where( c => c.Country == "Italy" );  
.OrderBy( c => c.Name )  
.Select( c => new { c.Name, c.City } );
```

Muchos constructores del lenguaje y sintaxis son necesarios para soportar lo que se ha visto hacer en pocas líneas de código de la consulta a datos. Bajo la cobertura de esta sencilla expresión de consulta está inferencia de tipos local, métodos de extensión, expresiones lambda, expresiones de inicialización de objetos, y tipos anónimos.

Todas estas características son útiles por si solas, pero en un cuadro global, se puede apreciar que hay un gran paso en dos direcciones: uno traslado a un estilo declarativo de codificación y una reducción de la impedancia de incompatibilidad entre código y código.

#### **2.18.6. LINQ to Objects**

LINQ to Objects tiene como objetivo manipular colecciones de objetos que pueden estar relacionados con otros de una forma jerárquica o un graph. Desde cierto punto de vista, LINQ to Objects es la implementación por default usada en una consulta LINQ, está disponible incluyendo el namespace System.Linq.

Las consultas LINQ to Objects no están limitadas por las colecciones generadas por los usuarios, como se puede ver con el siguiente ejemplo que muestra una consulta LINQ sobre información extraída del archivo de sistema. La lista de todos los archivos en un directorio dado es leída en memoria antes de ser filtrado por la consulta LINQ.

```
string tempPath = Path.GetTempPath();
DirectoryInfo dirInfo = new DirectoryInfo( tempPath );
var query =
from f in dirInfo.GetFiles()
where f.Length > 10000
orderby f.Length descending
select f;
```

#### **2.18.7. LINQ to ADO.NET**

LINQ to ADO.NET incluye diferentes implementaciones que comparten las necesidades para manipular datos relacionales, ello incluye otras tecnologías que son específicas para una capa de persistencia particular.

- **LINQ to SQL** Maneja el mapeo entre tipos custom en C# y el esquema de la tabla física.
- **LINQ to Entities** Es en varias formas similar a LINQ to SQL. Sin embargo, en lugar de usar la base de datos física como una capa de persistencia, utiliza un modelo conceptual de entidades de datos EDM (Entity Data Model). El resultado es una capa abstracta que es independiente de la capa física de datos.
- **LINQ to DataSet** LINQ to SQL y LINQ to Entities tienen características similares porque ambos acceden a datos almacenados en base de datos relacionales y operan sobre

entidades de objetos que representan a datos externos en memoria. La principal diferencia está en que ellos operan sobre un nivel diferente de abstracción, mientras que LINQ to SQL está atado a una estructura de base de datos física. LINQ to Entities opera sobre un modelo conceptual (entidades de negocio) que puede estar más lejos de una estructura física.

La razón para estas diferentes opciones para acceder a datos relacionales a través de LINQ es que los diferentes modelos para acceso a base de datos están siendo utilizados hoy en día. LINQ intenta ofrecer ayuda y mejoras en acceso a base de datos sin forzar la adopción de un modelo común.

## 2.19. Microsoft SQL Server 2008


SQL Server 2008 proporciona la tecnología y capacidades de las que dependen las organizaciones para administrar el creciente desafío de la administración de datos y entrega de información procesable a los usuarios. Con avances significativos en áreas clave, SQL Server 2008 es una parte integral de la visión de Plataforma de Datos de Microsoft, que es diseñada

para satisfacer las necesidades de administración y trabajo con datos.



Es un producto significativo que trae nuevas características y mejoras claves, siendo la entrega más robusta y completa de SQL Server hasta la fecha.

SQL Server incluye varias tecnologías de administración y análisis de datos.

 **Motor de Base de Datos.**- Es el servicio principal para almacenamiento, procesamiento y seguridad de datos. Database Engine proporciona acceso controlado y procesamiento rápido de transacciones enfrentando los requerimientos de la mayor demanda de aplicaciones consumidoras de datos del negocio.





**Integración de Servicios.-** Es una plataforma para la construcción de soluciones de integración de datos de alto rendimiento, incluyendo paquetes que proporcionan procesamiento de extracción, transformación y carga ETL (extract, transform, and load) desde almacenes de datos.



**Analysis Services – Datos multidimensionales.-** Analysis Services soporta OLAP permitiendo diseñar, crear y administrar estructuras que contienen datos agregados desde otras fuentes, como bases de datos relacionales.



**Replicación.-** Replication es un conjunto de tecnologías para copiar y distribuir datos y objetos de base de datos desde una base a otra, y consecuentemente la sincronización entre las bases de datos para mantener consistencia.



**Analysis Services – Minería de Datos.-** Analysis Services permite diseñar, crear y visualizar modelos de minería de datos. Estos modelos de minería pueden ser construidos desde otras fuentes por medio del uso de una amplia variedad de estándares de algoritmos de minería de datos.



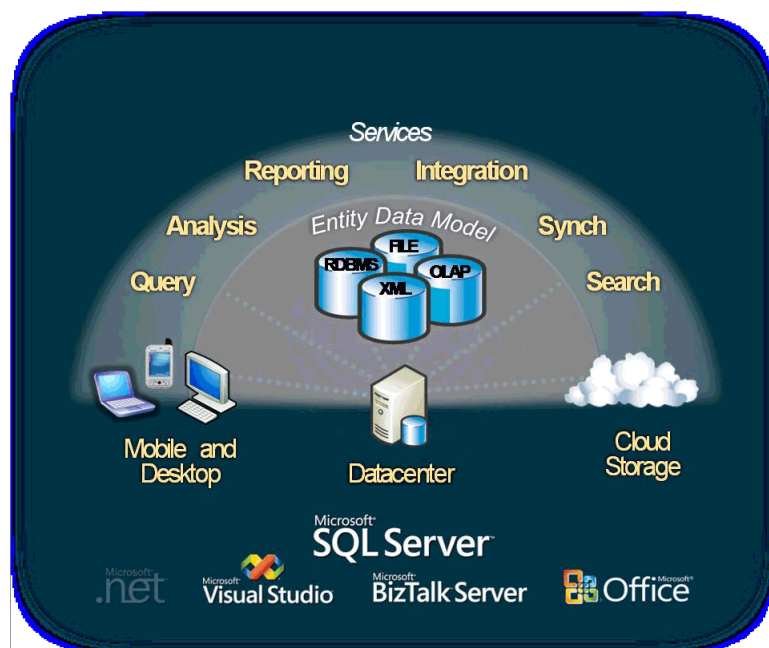
**SQL Server Service Broker.-** Services Broker ayuda a los desarrolladores a construir aplicaciones de base de datos escalables, y seguras. Esta nueva tecnología de ingeniería de base de datos proporciona una plataforma de comunicación basada en mensajes que permite a los componentes independientes de una aplicación ejecutarse como una funcionalidad completa.



**Reporting Services.-** Proporciona funcionalidad para presentación de informes empresariales, habilitados en la Web, se puede crear reportes que trazan el contenido desde una variedad de fuentes de datos, publicar los reportes en varios formatos y administrar de forma centralizada la seguridad y suscripciones.

### 2.19.1. Visión de plataforma de datos de Microsoft

El almacenamiento de información se está habilitando para nuevos tipos de información, tales como la digitalización de imágenes y videos, y el sensor de información de etiquetas RFID (Radio Frequency IDentification), la cantidad de información digital dentro de una organización esta multiplicándose.



**Figura X.** Visión de plataforma de datos de Microsoft SQL Server

El enfoque de la visión de plataforma de datos de Microsoft se enfoca en los siguientes aspectos:

**Confiable.** Permite a las organizaciones correr sus aplicaciones más críticas con altos niveles de seguridad, fiabilidad y escalabilidad.

**Productivo.** Permite reducir el tiempo y costo requerido para desarrollar y administrar la infraestructura de datos.

**Inteligente.** Proporciona una plataforma completa que entrega perspectivas e información donde los usuarios lo requieran.

## 2.20. Partners de Migración de Crystal a Reporting Services

La tendencia en manejo de soluciones de reportes con la aparición y mejoras de Microsoft Reporting Services es la migración de aplicaciones anteriores basadas en Business Objects Crystal Reports, varias empresas han reconocido el fuerte impacto de Microsoft Reporting Services haciendo de su actividad principal la prestación de servicios de migración como Partners de Microsoft (Microsoft, 2007). Entre las principales empresas se cita:

### ➤ 90 Degree Software

90 Degree Software ofrece un entorno innovador y enriquecido para auditoría de informes que acelera la migración de Crystal Reports a Microsoft Reporting Services definition language



(RDL). 90 Degree Software reconoció que Microsoft Reporting Services, RDL se ha convertido en un líder de la industria de formatos de informes, por ello 90 Degree Software es 100% compatible con Microsoft Reporting

Services.

### ➤ Blue Fox Software

Blue Fox Software, White Tiger, proporciona componentes del sistema modular del negocio, Business Intelligence y automatización de procesos a bajo precio.



Construido enteramente en .NET y SQL Server, White Tiger, utiliza tanto SQL Reporting Services como Crystal Reports, e incluye funciones

específicas que permiten una fácil migración entre los datos actuales a Reporting Services. Los nuevos servicios de Reporting, pueden ser incorporados sin problemas a medida a los usuarios hacen uso del sistema, sin que ellos interrumpan su flujo de trabajo. White Tiger fue construido para la integración con sistemas empresariales existentes para proporcionar amplia funcionalidad y mejores características.

### ➤ dataReference



dataReference Inc., Microsoft Gold Certified Partner, ha desarrollado un proceso para la migración de Crystal Reports hacia Microsoft SQL Server Reporting Services, llamado Report Migrator.

Report Migrator, un producto Microsoft Front Runner para SQL Server 2008, provee unos de los procesos de conversión más completos del mercado. Algunas de sus funciones claves son la conversión de report fields, tables, charts, crosstab/matrix, sub-reports, groups/sections, style/formatting, llamadas a procedimientos almacenados, ODBC connections, sentencias de consulta select, y header/footer.

➤ **RptToRdl.Com**



RptToRdl.Com reduce la cantidad de tiempo y dinero requeridos para convertir las plantillas de Crystal Report (.RPT) al Nuevo Microsoft's SQL Server Reporting Services basado en formato XML (.RDL). RptToRdl.Com puede convertir un 90% de los reportes en plantillas Crystal. RptToRdl recibe un archivo RPT, y devuelve el respectivo RDL en la mayoría de los casos en las 24 a 48 horas siguientes. Además, también el usuario tiene acceso gratuito a la licencia de auxiliar, una utilidad que lleva a través de los pasos necesarios para terminar con el resto de la conversión del informe. RptToRdl.Com proporciona una forma rápida y rentable a las necesidades de la migración de informes.

➤ **KTL Solutions**



KTL Solutions, Inc., Microsoft Gold Certified Partner, es un servicio completo de consultoría tecnológica especializada en la organización de Microsoft Dynamics <sup>TM</sup> GP, de soluciones personalizadas para los clientes de Microsoft Dynamics GP, y de las soluciones de business intelligence. La solución de conversión Crystal convierte rápidamente la información a Microsoft SQL Server Reporting Services information. A través de la solución, los usuarios pueden sacar provecho a la inversión que hicieron sobre Crystal Reports mientras que toman todas las ventajas del diseño basado en web de Microsoft SQL Server Reporting Services, ayudando a reducir el costo total de las

necesidades de inteligencia de negocios. La solución Crystal Converter es completamente manejada por el usuario y convierte la mayoría de los informes en cuestión de segundos.

➤ **Result Data**



Partner de Crystal Reports / Business Objects y Microsoft Gold Partner, Result Data está calificada para ayudar a planificar, ejecutar y apoyar migración BI reporting. La misión de Result Data desde 1996 ha sido ofrecer a los clientes una mejor comprensión de sus empresas y los mercados mediante la mejora de su capacidad para reunir, analizar y actuar sobre la información. El certificado Business Objects Microsoft consultants, ha ayudado a cientos de clientes a construir y aplicar depósitos de datos, almacenes de datos, elaboración de informes y soluciones de arquitectura de BI.

➤ **Avanade**



Avanade es el integrador de sistemas más grande del mundo dedicada a la plataforma de inteligencia de negocios de Microsoft, y que se centra en la aplicación de potentes soluciones de BI con SQL Server 2008, Reporting y Analysis Services, Integration Services, ProClarity, Business Scorecard Manager y las nuevas capacidades de BI en Office 2007. Avanade tiene un conjunto único de funciones que hacen posible una migración satisfactoria de Crystal Reports a SQL Server 2008 Reporting Services, en particular para las grandes organizaciones.

➤ **TCS**



Crystal Reports puede ser bastante complejo para una migración debido a una multitud de razones como diversos formatos de reportes, interfaces complejas, múltiples capas de reportes embebidos, etc. Un esfuerzo de migración manual puede ser demasiado largo y costoso. TCS RECAST es una herramienta basada en servicios que faculta una migración integral de reportes desarrollados en Crystal Reports (RPT) al formato SQL Reporting Services 2000/2005 (RDL).

➤ **Hitachi**



Hitachi Consulting, Microsoft Gold Partner, ganador del premio Microsoft Business Intelligence / Data Management Partner of the Year en 2006,

Es una empresa global de consultoría. La solución de migración de reportes de Hitachi Consulting, proporciona a las empresas una migración de Crystal Reports a SQL Server Reporting Services amplia y eficaz.

El enfoque de la migración se basa en el acelerador de conversión automática de informes, lo que automáticamente convierte Crystal Reports en los Servicios de Reportes RDL.

➤ **BAC**

La misión de BAC es ayudar a los clientes a conseguir una ventaja competitiva en el Mercado a través de la integración de los procesos del negocio con la tecnología correcta. La migración de Crystal por BAC permite a las organizaciones aprovechar plenamente de Microsoft SQL Server Reporting Services.

Los consultores experimentados ayudan a los equipos de desarrollo en la formulación de normas, el desarrollo de plantillas, y la aplicación de las mejores prácticas al tiempo que migran a Reporting Services. BAC tiene un historial probado en desarrollo de alto valor soluciones de Reporting Services. Estas incluyen el desarrollo de una solución de presentación de informes para Microsoft.

➤ **KeyPulse**



Desde 1999 KeyPulse tiene una exitosa trayectoria en el desarrollo de data warehousing y soluciones de Business Intelligence (BI). Keypulse

ha diseñado una solución pendiente de patente que permite convertir fácilmente una o varias plataformas de BI, incluyendo el contenido de Crystal, a Microsoft Reporting y Analysis Services. CUBIT™ (Utilidades de conversión de las tecnologías de Business Intelligence) facilita la entrega de contenidos convertidos de forma rápida y precisa, al mismo tiempo que minimiza el tiempo, los recursos y los riesgos asociados a las conversiones puramente manuales.

➤ **Sonata**



Sonata's BI Center of Excellence, se enfoca en la creación de soluciones de business Intelligence en Microsoft SQL Server 2008, BI sStack y la migración a SQL reporting services. Sonata's engine (crème) permite la migración de las actuales plataformas de información - ya sea de Crystal o BO Informes - a Microsoft SQL Server 2008 Reporting Services.

➤ **Wipro**



Wipro, Microsoft Gold Partner, tiene una práctica dedicada a la Integración de Datos. Un área central de esta práctica es Microsoft SQL Server Reporting Services (SSRS). La práctica con su experiencia comprende migraciones de informes de Crystal a SSRS, dada la complejidad de las mismas, incluye un Kit de evaluación de las Migraciones y asesoría en la ejecución de los proyectos.

➤ **plus-IT**



plus-IT es un próspero negocio de IT-Consulting, dirigido principalmente a las empresas medianas y grandes en los mercados de habla alemán. Sus expertos analizan los procesos de negocio centrados en la presentación de informes, análisis y flujos de trabajo scorecarding.

➤ **SDM**



Strategic Data Management (SDM) provee la automatización de procesos de negocios y soluciones de presentación de informes. Utiliza su experiencia y las mejores herramientas en el mercado para ayudar a los clientes en la conversión de sus actuales informes operativos de Crystal Reports a Microsoft's Reporting Services. SDM también mantiene una experiencia específica en la conversión de Crystal Reports que hace uso de SAP BW a Microsoft Reporting Services.

## **CAPITULO III: IMPLEMENTACIÓN DEL SOFTWARE: DYNAMIC REPORT GENERATOR.**

### **3.1. Introducción**

Una de las principales tendencias de los sistemas informáticos de gestión, es la incorporación de módulos que abarquen tecnologías, aplicaciones y prácticas para el almacenamiento, integración, análisis y presentación de información, todas en conjunto adoptan el nombre de inteligencia de negocios. En los esfuerzos para satisfacer estas necesidades, se han incluido un conjunto fragmentado de herramientas y sistemas que son costosos, pesados y difíciles de manejar y mantener, y a menudo demasiado especializados o intensivos en cuanto a recursos para servir a un amplio grupo de usuarios.

La generación de un informe, es sin duda un hito en el agitado mundo empresarial, a partir de la materialización del estado de una entidad en un espacio y tiempo delimitado, se hace indispensable para el análisis de diferentes ámbitos económicos, comerciales de gestión y administración.

Con el fin de aportar con una herramienta que constituya un método fácil y adaptativo de generación de informes, el presente trabajo contempla un análisis de la problemática y las alternativas de solución, el establecimiento del diseño arquitectónico y funcional, la selección de mecanismos y tecnología, y la definición de especificaciones técnicas de desarrollo de software basándose en la metodología del proceso unificado de software.



El proceso unificado de desarrollo de software establece como fases principales la Ingeniería de la Información, Análisis, Diseño, Implementación y Pruebas.

El caso práctico de la investigación, ha sido llevado a cabo en cuatro etapas:

- a) **Primera Etapa.**- Recolección y organización de la información. Consiste en la investigación del tema y la consiguiente evaluación de la información obtenida.
- b) **Segunda Etapa.**- Estudio global e identificación de requerimientos funcionales en los cuales se sustentará la investigación.
- c) **Tercera Etapa.**- Comprende las fases de análisis, diseño e implementación.
- d) **Cuarta Etapa.**- Incluye las fases de pruebas, documentación y presentación del trabajo investigativo. Así como la demostración práctica del módulo implementado.

Como entregables del caso práctico, constan:

**Tabla III-1.** Entregables DRG

<b>Entregable</b>	<b>Descripción</b>
<b>Instalador DRG Configurator</b>	Herramienta de configuración para acceso a datos y reportes.
<b>Merge Module DRG</b>	Merge module que la aplicación Anfitrión puede utilizar para generar su instalador.
<b>Archivo Excel generador de scripts de carga</b>	Un archivo Excel en donde consta la generación de scripts para carga de sinónimos acorde la estructura de la base de datos.
<b>Manual de usuario</b>	El manual instructivo para el uso de DRG.

### **3.1.1. Presentación**

El problema que se pretende resolver, surge a partir de la necesidad de los usuarios de crear reportes acoplados a sus requerimientos sin tener que estar sujetos a las restricciones de diseño, aprovechando al máximo los datos existentes en los repositorios, para toma de decisiones coherentes y previsoras.

La herramienta Dynamic Report Generator, hará posible que los usuarios generen sus propios reportes a petición, haciendo uso de mecanismos intuitivos de acceso a datos y aprovechando al máximo el capital más importante de una organización, la información.

Se hará uso de la herramienta con mejores características para generación de reportes ad-hoc, a partir del análisis comparativo entre las herramientas para la generación de informes, especialmente basados en la arquitectura, y librerías que provean métodos de programación para definición, construcción y publicación.

La experiencia adquirida en aplicaciones de presentación de reportes estáticos en donde durante la etapa de vida del sistema, se presentan variaciones significativas en los requerimientos de un informe, que se traducen en un incremento sustancial del costo del software y problemas de negociación inherentes a los mismos, movilización de recursos humanos y demoras en las entregas; ha conllevado a utilizar un enfoque de dinamismo orientado a construir una aplicación de calidad, haciendo énfasis en la reutilización, que se adapte a los cambios de requerimientos del cliente, y sea de fácil mantenimiento. Basados en el desarrollo de software de gestión con un modelo de proceso de software en espiral, que es el que se ha venido utilizando en el desarrollo del sistema matriz de Dynamic Report Generator, la Suite Mercado de Capitales.

Se pretende aprovechar al máximo las novedades incorporadas en el más reciente lanzamiento de Visual Studio .Net, como LINQ y características relacionadas.

### **3.1.2. Metodología de Desarrollo**

Se utilizará los principios del paradigma de desarrollo orientado a objetos, basado en el proceso unificado de desarrollo de software propuesto por Jacobson, Booch y Rumbaugh (RUP). Utilizando el lenguaje unificado de modelado (UML).

### **3.1.3. Visión**

Hacer de la herramienta Dynamic Report Generator, un componente adaptable a cualquier sistema web con una base relacional, siendo un producto comerciable de manera independiente y con un ciclo de vida extensible.

#### **3.1.4. Ambientes de Operación**

Los ambientes de operación del proyecto están delimitados por el software principal con el que Dynamic Report Generator interactuará, que básicamente constituyen aplicaciones distribuidas en un entorno web desarrollado para un conjunto limitado de usuarios en un entorno empresarial cerrado.

#### **3.1.5. Referencias**

- IEEE-STD-830-1998: ESPECIFICACIONES DE LOS REQUISITOS DEL SOFTWARE

### **3.2. Metas de Diseño**

Se ha definido las características operaciones de la infraestructura y sistema empresarial del proyecto, en base al sistema principal con el cuál interactuará la herramienta Dynamic Report Generator.

#### **3.2.1. Rendimiento**

A pesar de no existir requerimientos específicos de alto grado de rendimiento, es importante mantener niveles aceptables de respuesta de forma que en un ambiente altamente concurrente, el sistema responda eficientemente y proporcione los resultados esperados. Al desarrollar la herramienta DRG, como parte de una aplicación web, las páginas deberán ser lo más sencillas posibles, pero siempre, guardando fidelidad con la comprensión y facilidad de uso del usuario final.

#### **3.2.2. Disponibilidad**

La herramienta DRG, deberá estar disponible todo el tiempo, mediante la publicación en un servidor web operativo.

### **3.2.3. Fiabilidad**

Uno de los principales objetivos del presente trabajo, es la fiabilidad de los datos presentados al usuario final, es por ello que la herramienta DRG, debe permitir la creación de reportes altamente fiable.

### **3.2.4. Seguridad**

La herramienta DRG, no implementará medios de autenticación ni temas relacionados, este aspecto deberá ser administrado por las aplicaciones web contenedoras.

### **3.2.5. Interoperabilidad**

Uno de los principios básicos de la herramienta DRG, es la interoperabilidad con cualquier sistema de base de datos distribuida. Como punto de partida, DRG deberá interactuar con todos los sistemas de la Suite MDC.

## **3.3. Fase de Planificación**

### **3.3.1. Definición del ámbito del software**

Los clientes de sistemas de software de gestión, tienen la necesidad de obtener datos a la medida en el momento y lugar oportuno. La Suite MDC, siendo un software de gestión financiera, presenta como requerimiento el desarrollo de una herramienta para creación de reportes ad hoc. A pesar que en un principio la solución será utilizada en la Suite MDC, deberá ser diseñada para que cualquier aplicación la adopte como presentador de informes.

Al conseguir un buen producto, puede ser comercializable de manera independiente a clientes particulares como otras empresas de desarrollo en donde los costos de comprar un componente ya hecho son menores al desarrollo de soluciones que no necesariamente tendrán las ventajas de la herramienta RDG.

La herramienta Dynamic Report Generator, debe hacer posible que el usuario final genere informes con datos significativos a su área de trabajo de manera fácil e intuitiva, mediante la manipulación de instrumentos descriptivos y familiares a su entorno. RDG, también debe

permitir el almacenamiento de los reportes creados por el usuario final para su posterior consulta y formas de distribución.

En el reporte, se debe abrir la posibilidad de incluir funciones de agrupación y cálculos adicionales. Así como el establecimiento de filtrado.

Uno de los retos es utilizar la tecnología reciente de Visual Studio .Net, también existe el riesgo que las expectativas colocadas sobre la herramienta para generación de informes de reporting services no se cumplan en base a los cuadros comparativos.

### **3.3.2. Perspectiva de Dynamic Report Generator**

La funcionalidad Dynamic Report Generator complementa la funcionalidad operativa y de control existente en los módulos de negocio, como: Interbancarios, Repos, Títulos Valores, Renta Variable, Cambios Spot, y Cambios Forward de la Suite MDC. Adicionalmente, interactuará con los módulos de Seguridades, Clientes, Información de Mercado, y Parametrización.

Las principales entidades externas que intervendrán con el sistema son:

- Clientes
- Usuarios del Sistema
- Supervisores de Negocio

### **3.3.3. Características del Usuario**

El usuario que usará la funcionalidad de la aplicación deberá tener los siguientes conocimientos:

- Conocimientos elementales de informática
- Conocimientos básicos de navegación (Internet)
- Conocimiento de las operaciones y del negocio de tesorería de una empresa
- Conocimiento del Sistema Operativo Windows (es sus diferentes versiones)
- Conocimiento de Microsoft Office.

### 3.3.4. Perfiles de Usuario y Accesos al Sistema

**Tabla III-2.** Perfiles de Usuario y Accesos al Sistema DRG

Sección	Opción de Usuario	Administrador	Operador	Consulta	Destinatario
Inicio		X	X	X	X
Administración	Definición de variables	X			
Reportes	Administración de reportes Ad Hoc	X	X		
	Consulta de reportes	X	X	X	X
	Definición Mecanismos de Distribución	X	X		
	Recepción de reportes distribuidos	X	X	X	X

### 3.3.5. Especificación de requerimientos

A partir de un breve esbozo del ámbito del sistema, es posible particionar las especificaciones con un detalle ligeramente mayor, separando las especificaciones funcionales de los atributos del sistema.

#### 3.3.5.1. Especificación de Funciones del Sistema

- RDG debe satisfacer las necesidades de creación de reportes ad hoc, siendo un componente sirva de medio para el acceso y manipulación de datos de cualquier base relacional, la misma que permitirá una interacción directa entre el usuario y los elementos de la base de datos.
- La herramienta Dynamic Report Generator, de posibilitar al usuario final para que genere informes con datos significativos a su área de trabajo de manera fácil e intuitiva.
- Cada informe estará construido en base a mecanismos de acceso a la base de datos definidos previamente, en donde un usuario con conocimientos intermedios de base de datos relacionales y del modelo de la aplicación con la cual interactuará DRG, de ahora en adelante llamado administrador DRG, establecerá la ruta de acceso a los datos de interés del usuario, proporcionando un nombre comprensible y detalles

difíciles de comprender por un usuario final. A estos mecanismos de acceso se les denominará variables.

- Las variables serán almacenadas en una base de datos independiente, desde las cuales el administrador DRG podrá crear, eliminar o modificarlas.
- El usuario final podrá acceder a una pantalla en donde seleccionará las variables de interés que desea consten en su reporte. Luego de esta selección, una siguiente fase de creación del reporte, permitirá que el usuario, establezca la disposición de las variables en el reporte.
- En esta primera fase, el desarrollo de la herramienta se basará en la definición de reportes lineales, sin embargo se debe dejar abierta la posibilidad de incluir tratamiento de elementos más complejos como matrices y gráficas.
- En el reporte, se debe abrir la posibilidad de incluir funciones de agrupación y cálculos adicionales. Así como el establecimiento de las variables que serán filtros para el reporte.
- Se dará la posibilidad de obtener una vista previa del reporte y con la conformidad del usuario se debe permitir almacenar el reporte en un repositorio de datos, con un nombre descriptivo y notas adicionales. De forma que el usuario que construyó el reporte, pueda volver a consultarlo cuando lo requiera, desde otra pantalla de administración de reportes creados.
- Se deberá permitir establecer mecanismos de distribución de reportes. Dichos mecanismos también deberán ser almacenados en un repositorio de base de datos para que sean eliminados o modificados cuando el usuario lo requiera.

#### **3.3.5.2. Especificación de Atributos del Sistema**

- El sistema debe ser una aplicación web que pueda ser utilizada como un merge module a un sistema anfitrión.
- La herramienta para generación de informes será Reporting Services.

- La administración de informes generados deberá almacenarse en un repositorio de base de datos.
- El sistema debe ser desarrollado en la tecnología Visual Studio .Net para mantener compatibilidad con la aplicación anfitrión Suite MDC.
- Como parte de los resultados esperados, vale mencionar, la facilidad de uso y tiempos de respuesta tolerables.

### 3.3.6. Estimaciones

En la estimación del costo y el esfuerzo del software intervienen variables complejas como, el capital humano, variables técnicas, de entorno, políticas, etc. Que inciden directamente en el costo del proyecto y en el esfuerzo requerido para llevarlo a cabo.

#### 3.3.6.1. Métricas

Para determinar el costo del proyecto, se utilizará métricas orientadas a la función, las mismas que utilizan una medida de la funcionalidad entregada por la aplicación como valor de normalización.

Se ha identificado los siguientes ajustes de valores de complejidad.

**Tabla III-3.** Ajuste de valores de complejidad

Factor	Valor
Copias de seguridad y de recuperación	0
Comunicación de datos	5
Procesamiento distribuido	4
Rendimiento crítico	1
Entorno operativo existente	5
Entrada de datos interactiva	4
Transacciones de entrada en múltiples pantallas	4
Archivos maestros actualizados de forma interactiva	3
Complejidad de valores del dominio de información	4
Complejidad del procesamiento interno	4
Código diseñado para la reutilización	5
Conversión/instalación en diseño	1



Instalaciones múltiples	5
Aplicación diseñada para el cambio	5
<b>Total</b>	<b>50</b>

Los factores de han evaluado acorde a:

**Tabla III-4.** Valores de ajuste

Categoría	Valor
No influencia	0
Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

La siguiente tabla describe los puntos de función identificados, ver anexo.

**Tabla III-5.** Computación de métricas de puntos de funciones

Parámetro de medición	Cuenta		Factor de ponderación			=
			Simple	Medio	Complejo	
Número de entradas de usuario	4	x	3	4	<u>6</u>	24
Número de salidas de usuario	11	x	4	<u>5</u>	7	55
Número de peticiones de usuario	1	x	<u>3</u>	4	6	3
Número de archivos	1	x	<u>7</u>	10	15	7
Número de interfaces externas	1	x	5	<u>7</u>	10	7
<b>Cuenta total</b>						<b>96</b>

Para calcular los puntos de función, se aplica la siguiente relación:

$$PF = cuenta\_total \times \left[ 0,65 + 0,01 \times \sum_{i=1}^{14} Fi \right]$$

Donde *cuenta\_total* es la suma de todas las entradas obtenidas en la tabla  $F_i$  ( $i = 1$  a  $14$ ) son los valores de ajuste de complejidad de la tabla.

$$PF = 96 \times [0,65 + 0,01 \times 50] = 110$$

Una vez calculado el número de puntos de función, en base a información histórica de proyectos similares, se establece las relaciones:

- errores por PF
- defectos por PF
- \$ por PF
- Página de documentación por PF
- PF por persona-mes.

### 3.3.6.2. Costo total del desarrollo de la solución.

La productividad media para proyectos con características similares es 6,5 PF/pm. Dado que este proyecto es de carácter investigativo didáctico desarrollado por una persona, se omiten los cálculos de costo en función de PF.

Delimitando los costos de desarrollo en la planificación inicial:

**Tabla III-6. CU-03**

Recurso	Subtotal (USD)
Gastos de Movilización	500
Teléfono e Internet	80
Impresión y copias de documentos	150
Suministros de Oficina	150
Suministros de PC	40
Otros imprevistos	200
<b>TOTAL</b>	<b>1120</b>

Para los riesgos identificados sobre la línea de corte, se ha establecido un plan de reducción, supervisión y gestión de riesgo (RSGR), el mismo que consta en el Anexo 1.

### 3.3.7. Gestión de Riesgos

En el desarrollo de un proyecto software, se puede afrontar los riesgos de forma proactiva y reactiva. En el presente proyecto, se ha definido una tabla de riesgo, con los principales

riesgos identificados, a partir de la cual, se ha establecido los planes de reducción, supervisión y gestión de riesgo, en función de un ordenamiento por probabilidad de ocurrencia e impacto. En caso de que los riesgos no considerados ocurriesen, se adoptará una estrategia reactiva para tratarlos.

### 3.3.7.1. Identificación y proyección del riesgo

Se ha identificado los principales riesgos del proyecto respondiendo a las interrogantes sugeridas en las diferentes categorías de riesgo propuestas por Roger Pressman [PRESS98]. A los diferentes riesgos se ha asignado un porcentaje de ocurrencia **Incertidumbre** así como un índice del impacto en el proyecto en caso de que el riesgo se convierta en realidad **Pérdida**.

Como resultado se ha obtenido la siguiente tabla de riesgos:

**Tabla III-7.** Tabla de riesgos

Riesgo	Descripción	Categoría	Probabilidad	Impacto	RSGR
1	La planificación temporal puede verse afectada por factores externos al proyecto.	RP	70%	2	Anexo 1.a
2	El entorno de desarrollo a utilizar es nuevo y puede haber dificultad en la familiarización con el mismo	ED	50%	3	Anexo 1.b
3	El tamaño estimado puede ser demasiado pequeño con el tamaño real que se puede alcanzar	TP	25%	4	Anexo 1.c
4	No todos los usuarios finales se familiarizan con el uso del sistema	IN	20%	3	Anexo 1.d
Línea de corte de riesgo					
5	Ambigüedades en la especificación de requerimientos	RP	18%	2	
6	El numero de sistemas con el que interactúa la herramienta de generación de reportes, puede ocasionar complejidad en la administración de fuentes de acceso	IN	10%	2	
7	La tecnología no alcanzara las expectativas	RT	10%	2	

8	Los requisitos del producto una interfaz de usuario especial	RT	10%	3
9	Número de clientes que usarán este producto y la consistencia de sus necesidades relativas al producto	RC	10%	3
10	La herramienta de generación de reportes no se ajusta a los requerimientos de mercado	RM	5%	1

**Tabla III-8. Categoría de riesgos**

Categoría	Sigla
Riesgos del tamaño del producto	TP
Riesgos del impacto en el negocio	IN
Riesgos relacionados con el cliente	RC
Riesgos del proceso	RP
Riesgos tecnológicos	RT
Riesgos del entorno de desarrollo	ED
Riesgos asociados con el tamaño de la plantilla del personal y su experiencia	PP
Riesgo de Mercado	RM

**Tabla III-9. Impacto del riesgo**

Categoría	Impacto
Catastrófico	1
Crítico	2
Marginal	3
Despreciable	4

### 3.3.7.2. Evaluación del Riesgo

En base a la tabla de riesgos se ha establecido un conjunto de ternas de la forma:

$[r_i, l_i, x_i]$  donde  $r_i$  es el riesgo,  $l_i$  es la probabilidad del riesgo y  $x_i$  es el impacto del riesgo.

Teniendo:

$([1, 70, 2], [2, 50, 3], [3, 25, 4], [4, 20, 3])$

Se tomará especial atención con la ocurrencia de los riesgos 1 y 2, debido a que pueden afectar directamente sobre los tiempos del proyecto.

### 3.3.7.3. Gestión y supervisión del riesgo

Para los riesgos identificados sobre la línea de corte, se ha establecido un plan de reducción, supervisión y gestión de riesgo (RSGR), el mismo que consta en el Anexo 1.

## 3.4. Fase de Análisis

Siguiendo la metodología de desarrollo ha llevado a cabo un análisis orientado a objetos, basados en el proceso unificado y manteniendo el estándar UML para diagramación.

### 3.4.1. Definición de Casos de uso

Los casos de uso facilitan la descripción funcional del sistema, los diferentes escenarios descritos en la sección 3.3.2, son traducidos a los casos de uso o utilización que tendrá el sistema propuesto. En forma general, un caso de uso es una descripción escrita del papel que tendrá un actor al ocurrir la interacción con el sistema.

#### 3.4.1.1. Identificación de actores

De forma global, DRG, tendrá un limitado conjunto de actores primarios con los cuales existirá interacción, esto claro, no significa que el número de actores sea el número equivalente a usuarios que tendrá el sistema, el cual sin duda, será equivalente a los usuarios del sistema matriz.

**Tabla III-10.** Identificación actores primarios

Nombre	Descripción
Administrador DRG	Administra las variables de acceso a datos
Operador DRG	Crea y administra informes ad hoc
Consulta DRG	Consulta los informes disponibles
Destinatario DRG	Recibe los informes a los que está suscrito
Framework SuiteMDC	Aplicación contenedora de DRG

**3.4.1.2. Caso de uso general**

**Tabla III-11. Caso de uso general**

<b>Nombre</b>	<b>CU-01</b>	
<b>Caso de Uso</b>	Generar informes ad hoc	
<b>Actores</b>	Framework SuiteMDC, Administrador DRG, Operador DRG, Consulta DRG, Destinatario DRG	
<b>Propósito</b>	Permitir la generación de informes ad hoc por parte del usuario final (Operador) y distribuirlo mediando los mecanismos especificados.	
<b>Visión general</b>	Funcionalidad de la herramienta para generación de reportes dinámicos	
<b>Tipo</b>	Primario y esencial	
<b>Curso Típico de Eventos</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El Administrador DRG ingresa al sistema anfitrión Framework SuiteMDC y selecciona la opción DRG.	2. Presenta pantalla de administración de variables.	
3. El Administrador DRG selecciona la opción de creación de variables.	4. Solicita la información requerida y almacena los datos en el repositorio.	
5. El Operador DRG ingresa al sistema anfitrión Framework SuiteMDC y selecciona la opción DRG.	6. Muestra las variables existentes.	
7. El Operador DRG ingresa las especificaciones del reporte a crear.	8. Valida las entradas, carga el informe en el servidor de reportes y presenta vista previa.	
9. El Operador DRG da su conformidad para el informe y solicita el almacenamiento.	10. Graba el informe en el repositorio.	
11. El Operador DRG, establece mecanismos de distribución para el reporte creado.	12. Almacena los mecanismos de distribución.  13. Hace efectiva la distribución.	
14. El Destinatario DRG recibe el informe.		

3.4.1.3. Casos de uso particulares

Tabla III-12. CU-02

<b>Nombre</b>	<b>CU-02</b>	
<b>Caso de Uso</b>	Definir variables	
<b>Actores</b>	Framework SuiteMDC, Administrador DRG	
<b>Propósito</b>	Especificar la forma en la que el administrador de la herramienta de generación de reportes establecerá el mecanismo de acceso a los datos.	
<b>Visión general</b>	El Administrador DRG ingresa las variables de acceso a los datos que serán utilizados luego por el operador DRG	
<b>Tipo</b>	Primario - Escencial	
<b>Post condiciones</b>	Se podrá ejecutar CU-03 (Crear reportes Ad Hoc)	
<b>Curso Típico de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El Administrador DRG ingresa al sistema anfitrión Framework SuiteMDC y selecciona la opción DRG.	2. Presenta pantalla de administración de variables y presenta menú de opciones.
3.	El Administrador DRG selecciona la opción de creación de variables.	4. Solicita datos descriptivos de la nueva variable (Nombre, descripción, tabla destino, campo destino).
5.	El Administrador DRG ingresa los datos solicitados.	6. Solicita el ingreso de la ruta de acceso que consiste en las relaciones entre una tabla de origen (maestro) hacia la tabla destino especificada en el punto 4.
7.	El Administrador DRG ingresa la ruta de acceso.	8. Valida que posibles inconsistencias en la definición de la ruta.
8.	Selecciona la opción guardar.	10 Almacena la variable en la base de datos.
<b>Cursos Alternativos</b>		
Línea 3.	El Administrador selecciona una opción distinta (eliminar, o consultar). El sistema realiza la acción solicitada.	
Línea 8.	En caso de inconsistencias presenta mensaje al usuario para que pueda corregir.	

Tabla III-13. CU-03

<b>Nombre</b>	<b>CU-03</b>
---------------	--------------

<b>Caso de Uso</b>	Crear Reportes Ad Hoc	
<b>Actores</b>	Framework SuiteMDC, Operador DRG	
<b>Propósito</b>	Especificar la forma en la que el operador de la herramienta creará un reporte ad hoc.	
<b>Visión general</b>	El operador DRG ingresa a la pantalla de creación de reportes, selecciona las variables a que va a contener el mismo y delimita la ubicación de cada variable en el informe.	
<b>Tipo</b>	Primario – Escencial	
<b>Pre condiciones</b>	CU-02 (Definir Variables)	
<b>Post condiciones</b>	CU-04 (Mecanismos de Distribución), CU-05 (Consultar Reportes)	
<b>Curso Típico de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
	El Operador DRG ingresa al sistema anfitrión	
1.	Framework SuiteMDC y selecciona la opción DRG.	2. Presenta un listado de variables disponibles.
3.	Selecciona las variables que conformarán el reporte	4. Muestra el listado de elementos que contendrá el reporte (Tabla, matriz, imagen).
5.	Selecciona los elementos que contendrá el reporte.	5. Muestra una un listado de las variables seleccionadas y otro listado de los elementos que contendrá el reporte.
7.	Distribuye las variables en las diferentes secciones de los elementos que apliquen.	8, Valida la ubicación. Y solicita detalles de agrupación y filtros.
8.	Ingresas los datos de agrupación y filtros.	9. Crea la definición del reporte y lo publica en el servidor.
10	Solicita vista previa.	11 Presenta vista previa
12	Solicita grabar.	13 Solicita datos descriptivos del reporte.
14	Ingresas los datos descriptivos del reporte	15 Graba el reporte en la base de datos.
<b>Cursos Alternativos</b>		
Línea 9.	En caso de presentarse errores en la definición del reporte se notificará al usuario para la redefinición de los elementos del mismo	
Línea 12	El Operador DRG puede solamente requerir consultar el reporte y no almacenarlo.	

**Tabla III-14. CU-04**

<b>Nombre</b>	<b>CU-04</b>
---------------	--------------



<b>Caso de Uso</b>	Definir de mecanismos de distribución	
<b>Actores</b>	Framework SuiteMDC, Operador DRG, Destinatario DRG	
<b>Propósito</b>	Especificar las formas de distribución de los reportes existentes.	
<b>Visión general</b>	El operador DRG ingresa a la pantalla donde se listan los reportes previamente creados, selecciona uno y asocia un mecanismo de distribución por mail o ubicación en una ruta específica.	
<b>Tipo</b>	Secundario – Escencial	
<b>Pre condiciones</b>	CU-03 (Crear Reportes Ad Hoc)	
<b>Post condiciones</b>	El Destinatario DRG recibirá el informe de acuerdo a los establecido en la definición del mecanismo de distribución sin necesidad de una nueva intervención del Operador DRG	
<b>Curso Típico de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
	El Operador DRG ingresa al sistema anfitrión	
1.	Framework SuiteMDC y selecciona la opción distribución DRG.	2. Presenta un listado reportes disponibles.
3.	Selecciona un reporte.	4. Presenta los mecanismos de distribución asociados al reporte.
5.	Selecciona una acción DIU a ejecutar sobre los mecanismos de distribución asociados al reporte.	5. Solicita la información requerida para cada acción, en caso de que la acción sea un ingreso solicitará los parámetros propios de cada mecanismo de distribución.
7.	Ingresa la información solicitada.	8. Realiza la publicación de la información ingresada por el usuario.
8.	El Destinatario DRG recibe el informe distribuido.	

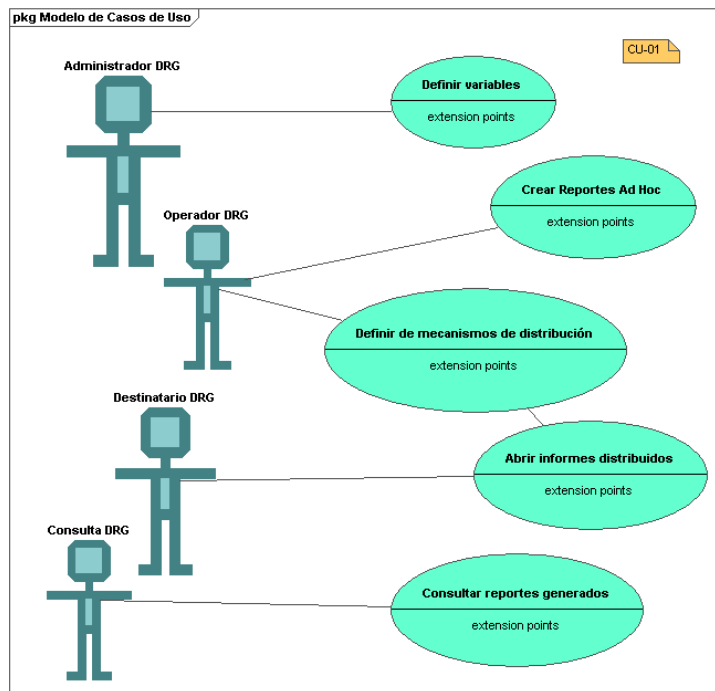
**Tabla III-15. CU-05**

<b>Nombre</b>	<b>CU-05</b>
<b>Caso de Uso</b>	Consultar reportes generados
<b>Actores</b>	Framework SuiteMDC, Consulta DRG
<b>Propósito</b>	Definir la forma de consulta de los reportes generados por el operador DRG

<b>Visión general</b>	El usuario de consulta DRG ingresa a la pantalla donde se listan los reportes previamente creados, selecciona uno y solicita la vista previa.	
<b>Tipo</b>	Primario – Escencial	
<b>Pre condiciones</b>	CU-03 (Crear Reportes Ad Hoc)	
<b>Curso Típico de Eventos</b>		
<b>Actor</b>		<b>Sistema</b>
El usuario de Consulta DRG ingresa al sistema anfitrión Framework SuiteMDC y selecciona la opción consulta DRG.	1.	2. Presenta un listado reportes disponibles.
3. Selecciona un reporte. Y solicita vista previa.	3.	4. Presenta la vista previa del informe.

### 3.4.2. Diagramas de Casos de uso

#### 3.4.2.1. Diagrama de caso de uso general – Contexto Funcional



**Figura XI.** Diagrama de caso de uso general

**Descripción:**

En este diagrama se muestra de modo general las principales interacciones que el sistema DRG llevará a cabo, se identifica los usuarios del sistema.

### 3.4.2.2. Diagramas de caso de uso particulares

#### CU-02 Definir variables

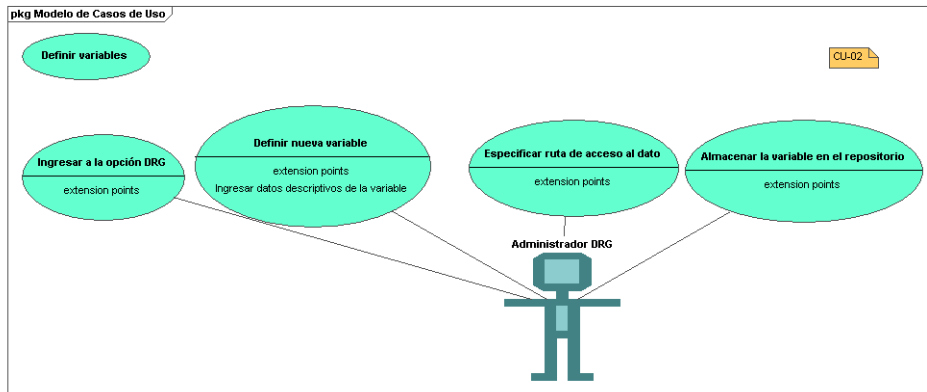


Figura XII. CU-02 Definir variables

#### Descripción:

En este diagrama se muestra la secuencia de ingreso de una variable, elemento constituyente de la solución DRG.

#### CU-03 Crear Reportes Ad Hoc

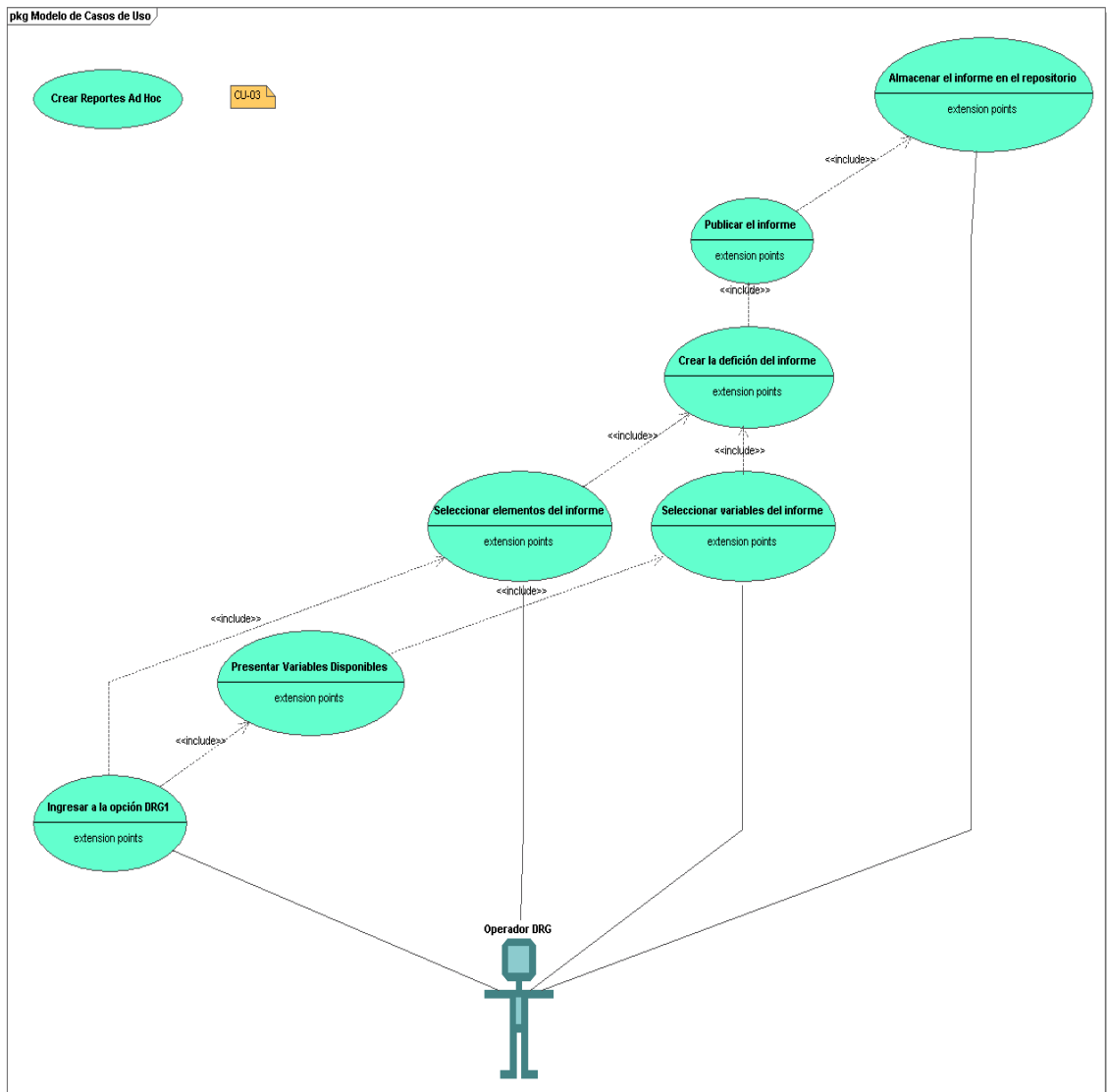


Figura XIII. CU-03 Crear Reportes Ad Hoc

### Descripción

En este diagrama se muestra la funcionalidad relacionada con la creación y publicación de los reportes ad hoc.

### CU-04 Definir de mecanismos de distribución

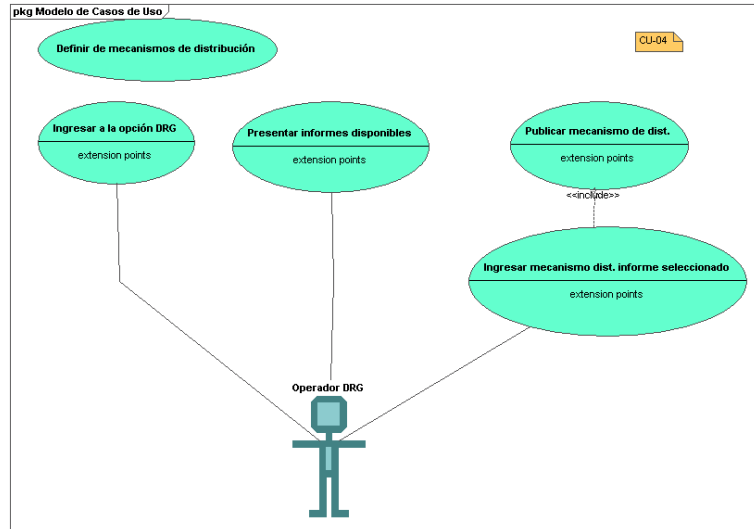


Figura XIV. CU-04 Definir de mecanismos de distribución

#### Descripción

En este diagrama se muestra la funcionalidad requerida para la creación de mecanismos de distribución de los reportes.

### CU-05 Consultar reportes generados

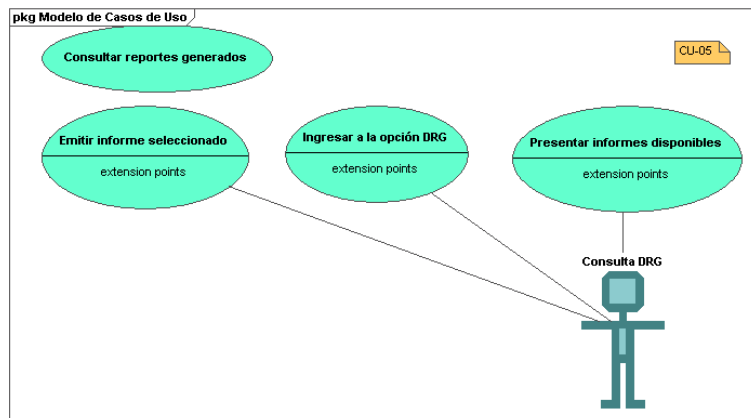


Figura XV. CU-05 Consultar reportes generados

#### Descripción

Este diagrama muestra la forma en la que el usuario de consulta podrá acceder a los reportes existentes.

### 3.4.3. Construcción del Modelo Conceptual

#### 3.4.3.1. Diagrama conceptual de base de datos

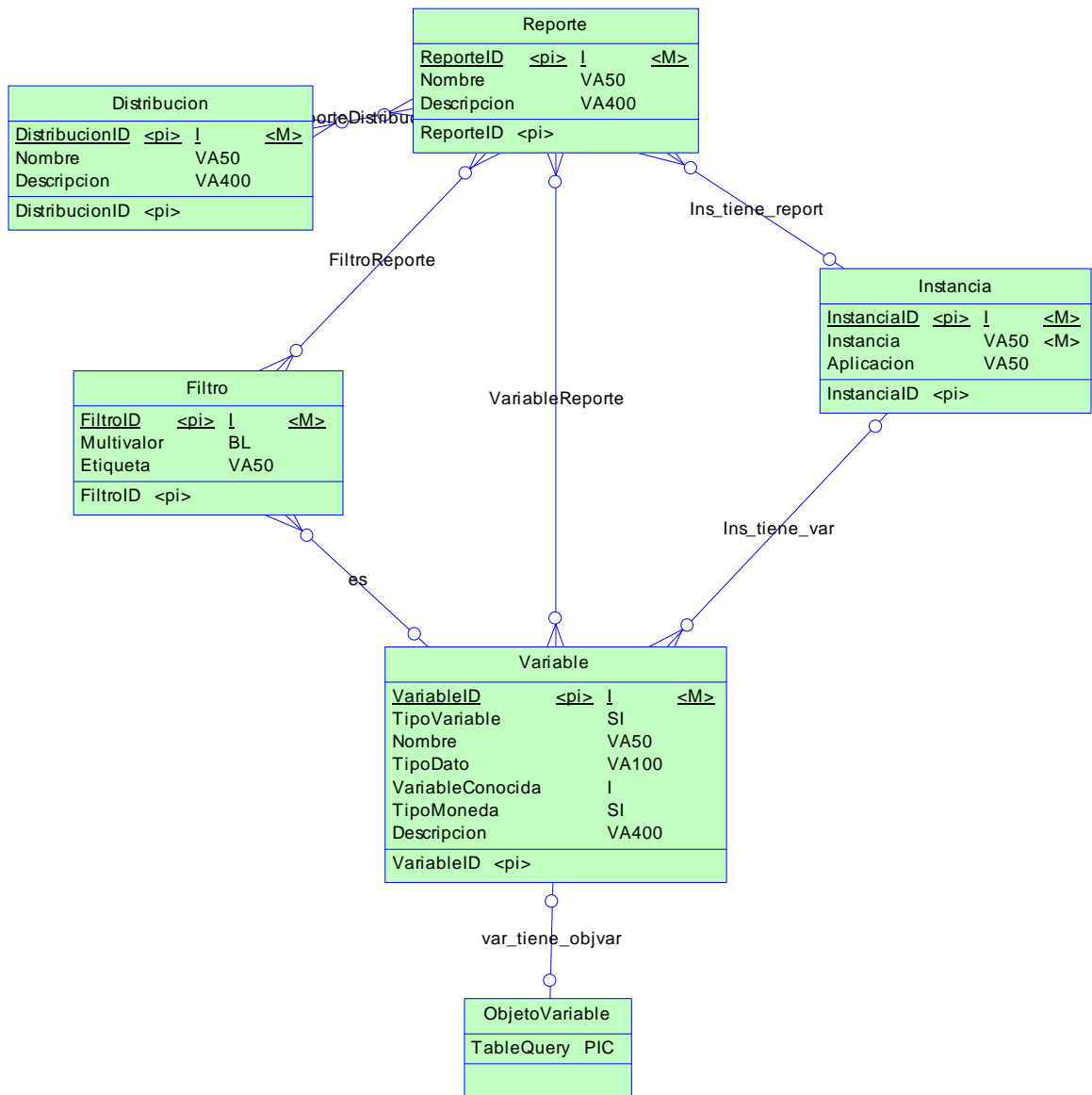


Figura XVI. Diagrama conceptual de base de datos

Descripción:

En este diagrama se muestra un modelado de las entidades que conformaran la base de datos y sus principales relaciones.

### 3.4.4. Construcción del Modelo Lógico

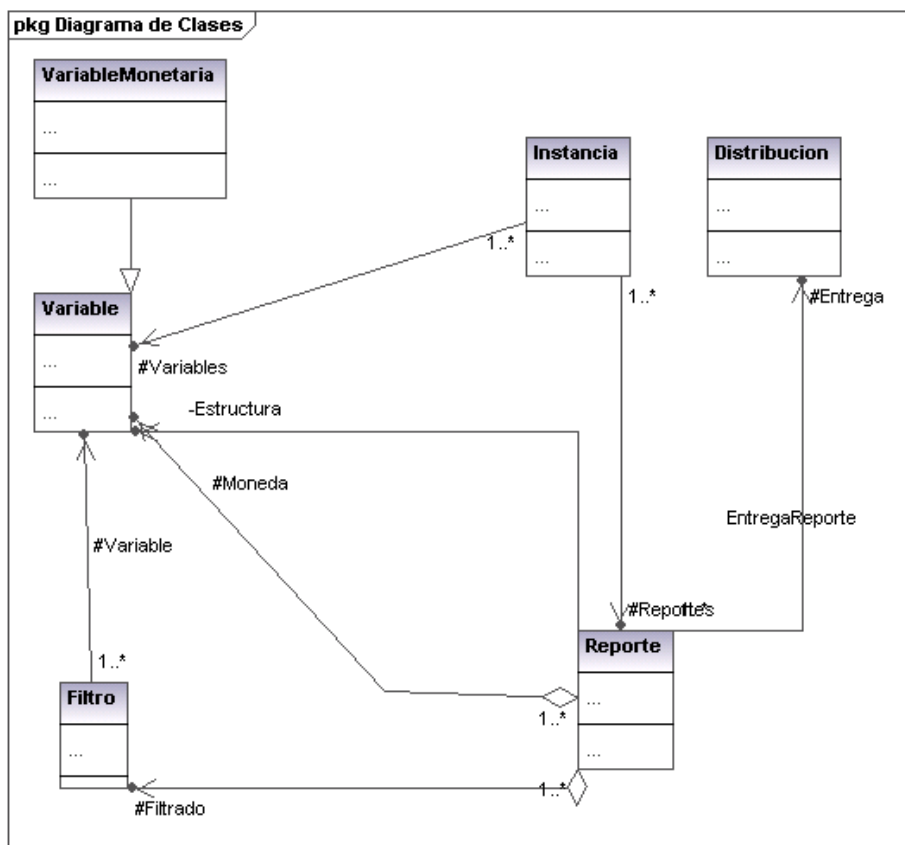


Figura XVII. Diagrama de Clases

**Descripción:**

En este diagrama se muestra las relaciones existentes entre las principales clases identificadas.



### 3.4.5. Diagrama de Estados

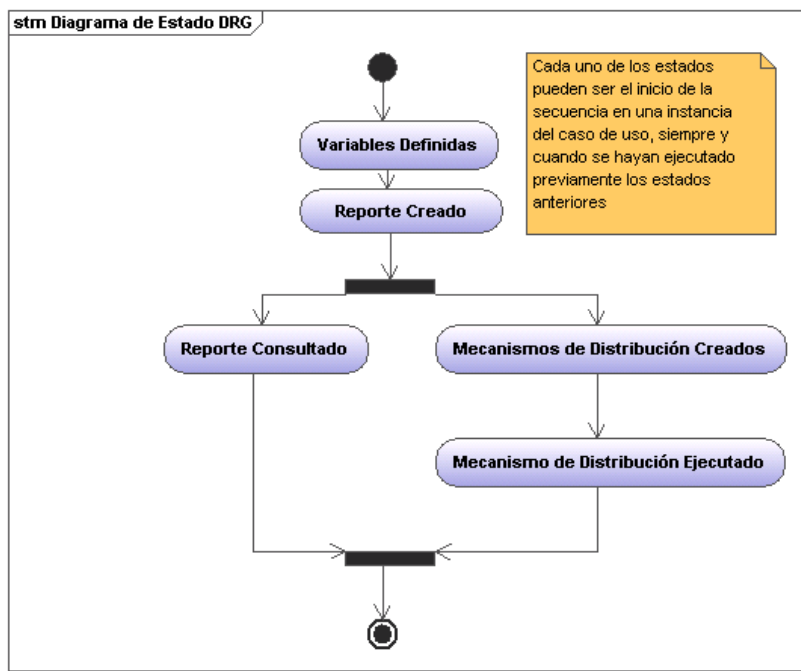


Figura XVIII. Diagrama de Estados

**Descripción:**

En este diagrama se muestra las principales transiciones en la vida del objeto reporte, elemento principal de la solución DRG.

### 3.4.6. Diagramas de Secuencia

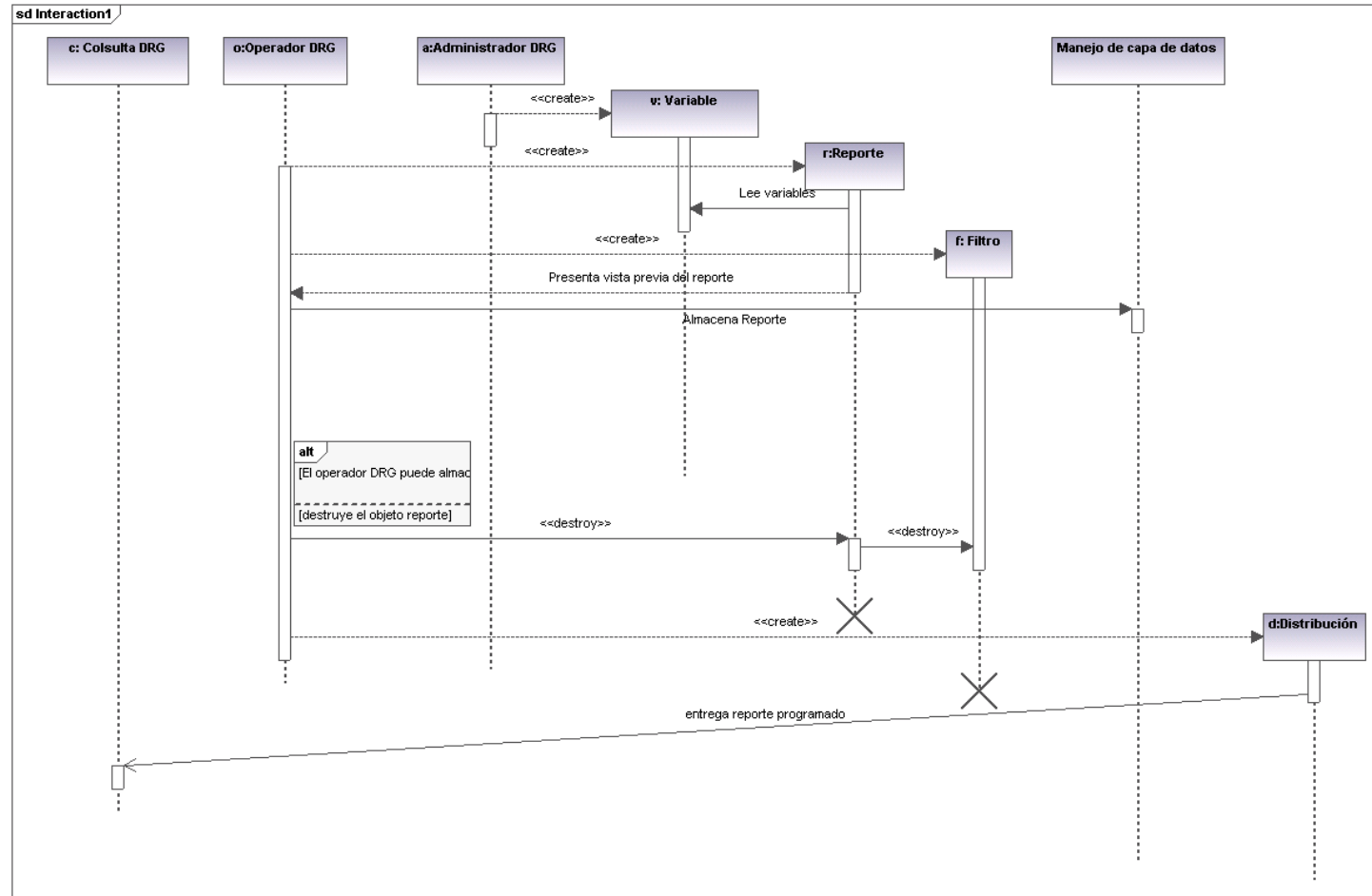


Figura XIX. Diagrama de Secuencia

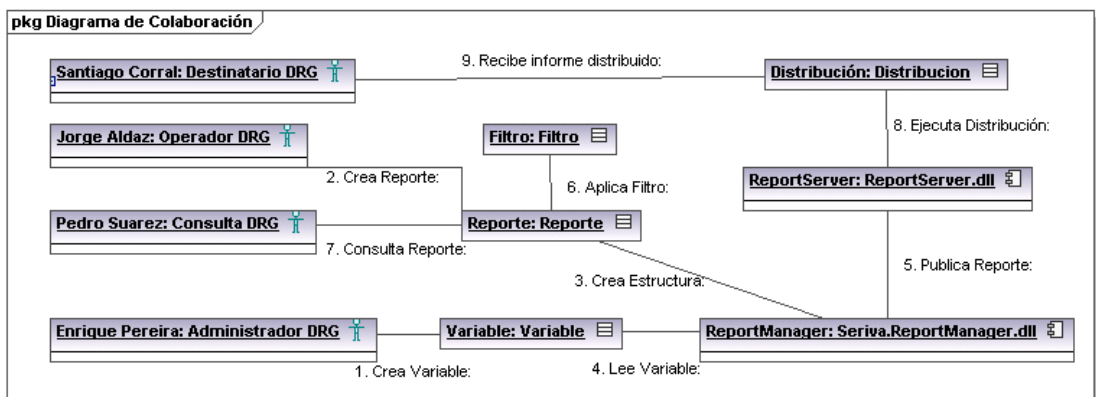
**Descripción:**

En este diagrama se muestra de modo secuencial la línea de vida de los objetos durante las diferentes fases de generación de reportes ad hoc

**3.5. Fase de Diseño**

El diseño se constituye en una fase de modelación del sistema, encontrando su forma incluido la arquitectura para que soporte los requisitos, tomando como entrada el resultado del análisis.

**3.5.1. Diagrama de Colaboración**



**Figura XX.** Diagrama de Colaboración

**Descripción:**

En este diagrama se muestra las principales interacciones entre los objetos organizado entre los objetos y los enlaces entre ellos.

### 3.5.2. Diagrama de Clases del Diseño

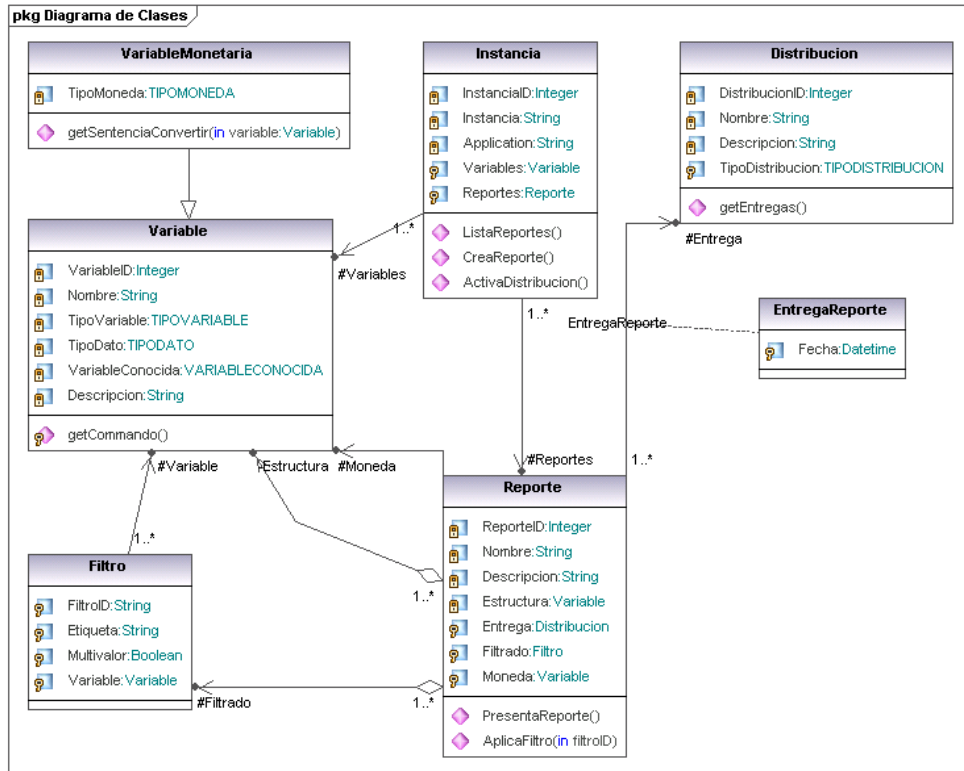


Figura XXI. Diagrama de clases del diseño

#### Descripción:

En este diagrama se muestra las principales clases que conformarán el sistema DRG, únicamente se centra en el modulo de generación de reportes y se omite las clases de los módulos relacionados.



### 3.5.4. Diagrama de Componentes

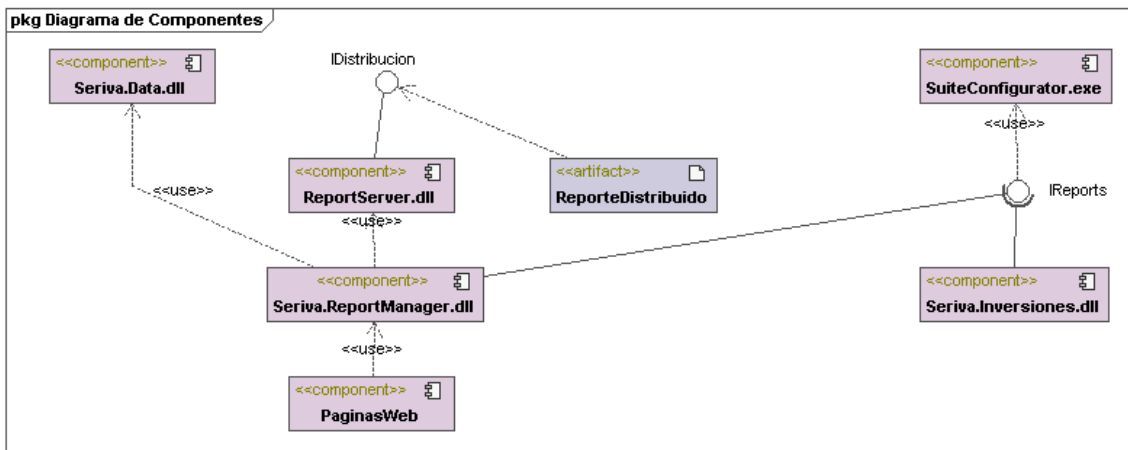


Figura XXIII. Diagrama de componentes

#### Descripción:

En este diagrama se muestra los componentes físicos del sistema.

### 3.5.5. Diagrama de Despliegue

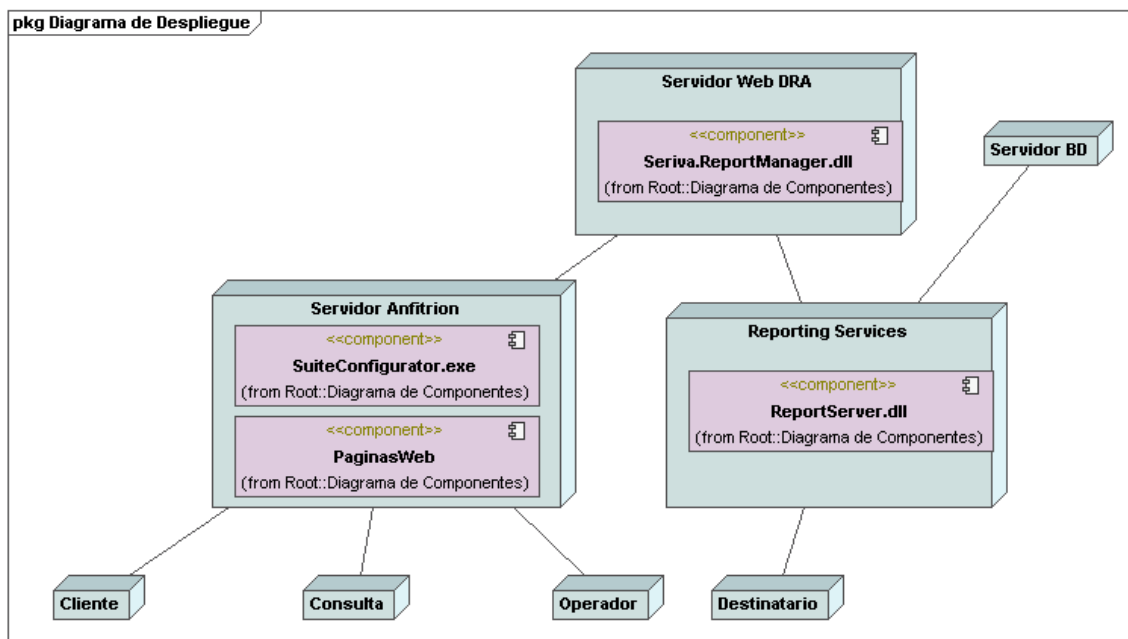


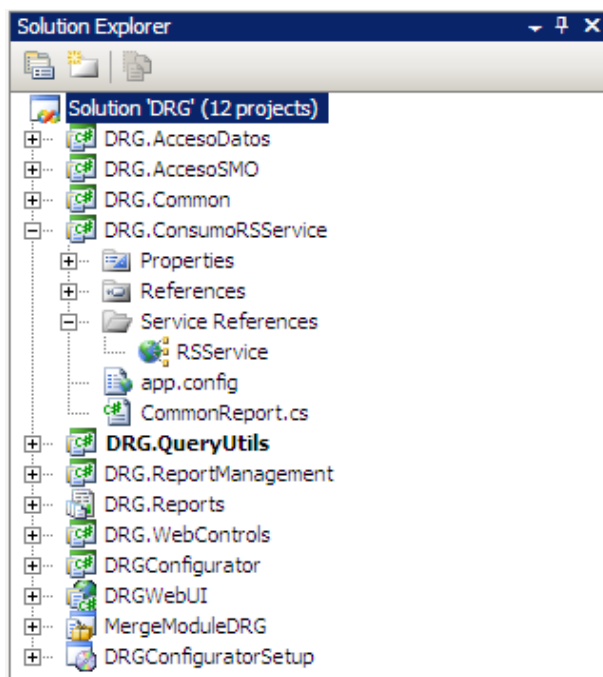
Figura XXIV. Diagrama de despliegue

**Descripción:**

En este diagrama se muestra una esquematización de la distribución de los componentes en los nodos que lo alojaran.

**3.6. Fase de Implementación**

Para la implementación de DRG se creó una solución en Visual Studio con la siguiente distribución de proyectos.



**Figura XXV.** Distribución de la solución DRG

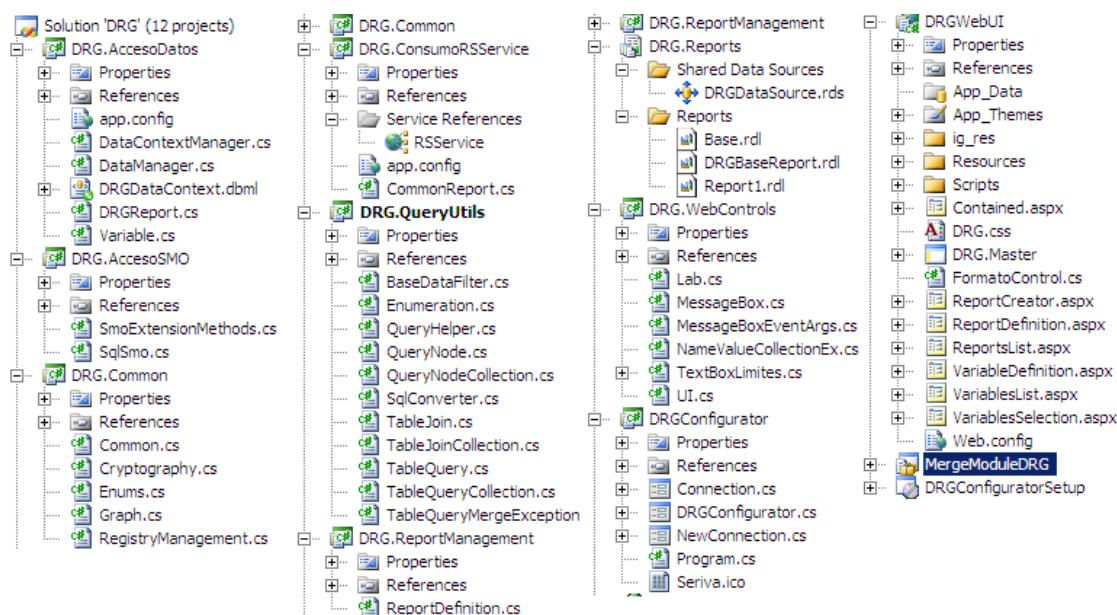
La capa de datos se compone de clases para el manejo de LINQ to SQL Classes, así como el manejo de `Microsoft.SqlServer.Management.Smo` para el descubrimiento de instancias de SQL server en la red y consulta de objetos de la base de datos.

Se aplicó el modelo de programación de LINQ para acceso a los datos de la base de DRG. Así como la teoría de grafos para el descubrimiento de relaciones entre los objetos de la base de datos del sistema anfitrión. Para la escritura de los archivos RDL se ha hecho uso del nuevo paradigma de manipulación XML presentado por Visual Studio 2008, LINQ to XML.

Como se ha indicado antes, la plataforma elegida para elaboración de informes ha sido Microsoft SQL Reporting Services, en la solución DRG también consta un proyecto de reportes bajo esta plataforma.

La siguiente figura muestra un detalle de los elementos de la solución DRG.





**Figura XXVI.** Elementos de la solución DRG

Los proyectos MergeModuleDRG y DRGConfiguratorSetup constituyen dos de los entregables del presente trabajo. El primero es la aplicación web que será embebida por un sistema anfitrión, y la segunda el instalador del DRG Configurator que permitirá establecer los mecanismos de acceso tanto a bases de datos como a reporting services.

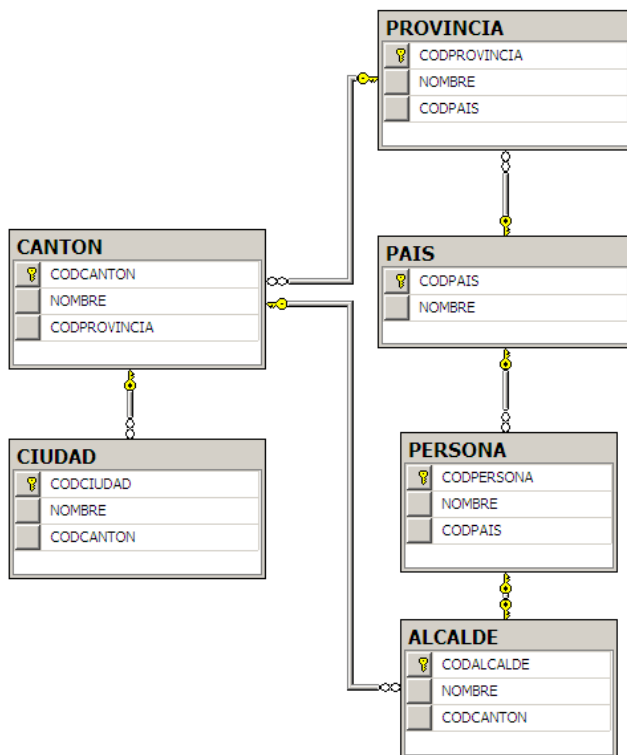
La salida principal del programa constituye la creación de reportes programáticamente, dado que reporting services se basa en el esquema XML, los archivos resultantes tienen la estructura del Anexo 2.

### 3.7. Fase de Estabilización

#### 3.7.1. Prueba Piloto

En esta etapa, se prueba la aplicación con un participante que puede tener menos requerimientos que en un test definitivo. La prueba piloto sirve para verificar que todo funciona como corresponde. Al finalizar se hace un pequeño análisis para determinar si es necesario hacer modificaciones.

En la prueba piloto para la herramienta DRG se ha utilizado una pequeña base de datos de prueba para establecer el correcto funcionamiento del aplicativo.



**Figura XXVII.** Base de datos de prueba de estabilización

Un ejemplo de la definición de una variable es el que sigue, en donde se define una variable para la tabla Alcalde.

La interfaz de usuario muestra los siguientes campos:

- Paso 1** | **Paso 2**
- Acceso a la Variable**
- Nombre:** Variable Alcalde
- Descripción:** Alcalde de un pais
- Origen:**
  - Instancia: UBICACIONINSTANCIA
  - Tabla: Pais
  - Aplicación: UBICACIONAPLICACION
- Destino:**
  - Instancia: UBICACIONINSTANCIA
  - Tabla: Alcalde
  - Columna: Nombre Alcalde
  - Aplicación: UBICACIONAPLICACION
- Aceptar**

**Figura XXVIII.** Datos generales de una variable

El usuario final únicamente requiere dar clic a la tabla de origen y la tabla de fin para que el aplicativo infiera el acceso a la misma.

Paso 1 Paso 2

Número de caminos 2

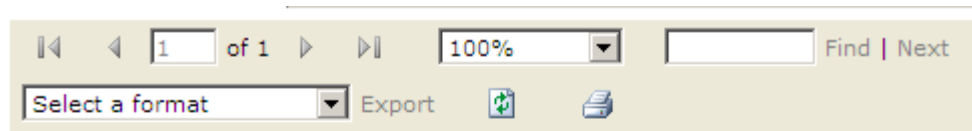
Camino Actual 2

- [-] Pais
  - [-] Codigo Pais - Codigo Pais
    - [-] Provincia
      - [-] Codigo Provincia Canton - Codigo Provincia Canton
        - [-] Canton
          - [-] Codigo Canton Alcalde - Codigo Canton Alcalde
            - Alcalde

Anterior Siguiente Grabar

**Figura XXIX.** Definición de esquema de acceso para una variable

La construcción del reporte para esta única variable se visualiza como sigue:



## Alcaldes de un Pais

Variable Alcalde
PACO MONCAYO
ANGEL YANEZ
GUSTAVO BAROJA
JAIME NEBOT

**Figura XXX.** Reporte generado.

## **CAPITULO IV: PRESENTACIÓN Y ANÁLISIS DE RESULTADOS**

### **4.1. Comprobación de la Hipótesis**

La hipótesis del presente estudio planteó que:

“Las características de la plataforma utilizada para la elaboración de informes en una aplicación web inciden directamente sobre la capacidad de definición y creación de reportes.”

A partir de esta información se puede identificar las variables dependientes e independientes que intervienen.

h1 = “Las características de la plataforma utilizada para la elaboración de informes en una aplicación web inciden directamente sobre la capacidad de definición y creación de reportes.”

h0 = “Las características de la plataforma utilizada para la elaboración de informes en una aplicación web NO inciden directamente sobre la capacidad de definición y creación de reportes.”

#### **4.1.1. Variables e indicadores**

**Variable independiente:** Características de la plataforma para la elaboración de informes.

**Variable dependiente:** Capacidad de definición y creación de reportes.

**Tabla IV-1.** Variables de la Hipótesis

Variable	Tipo	Concepto
Características de la plataforma para la elaboración de informes.	<ul style="list-style-type: none"><li>• Independiente</li><li>• Compleja</li></ul>	Toda aquella funcionalidad, arquitectura, diseño, herramientas, asistentes, etc. Que una plataforma para la elaboración de informes ofrece como parte de un producto distribuible para tales fines.
Capacidad de definición y creación de reportes.	<ul style="list-style-type: none"><li>• Dependiente</li><li>• Compleja</li></ul>	Idoneidad, competitividad, destrezas y habilidades que tanto el desarrollador de informes como los usuarios finales, disponen para definir estructuras de reportes o crear informes acorde con los requerimientos explícitos e implícitos de un sistema informático.

### Indicadores

Se ha adoptado como indicadores los elementos citados en las secciones 2.2 y 2.11 así como las valorizaciones de la sección 2.12, que establece los elementos que permiten establecer mediciones tanto de la variable independiente como de la variable dependiente.

4.1.2. Operacionalización Metodológica

Tabla IV-2. Operacionalización Metodológica de las Variables

Hipótesis	Variable	Indicadores	Técnica	Fuente de Verificación
<p>Las características de la plataforma utilizada para la elaboración de informes en una aplicación web inciden directamente sobre la capacidad de definición y creación de reportes.</p>	Características de la plataforma para la elaboración de informes.	Producto Arquitectura Estructura del Reporte Servidor de Reportes Herramientas Integración Manejo de Seguridad Visor de reportes Conectividad a Bases de Datos Características para construcción de reportes Características para desarrollo de reportes Tipos de reportes Ordenamiento y Agrupación Formato Objetos de reportes Parámetros Tipos de campos de la base de datos Explorador de Objetos Apertura para personalización	<ul style="list-style-type: none"> <li>•Revisión de documentación de los productos de estudio.</li> <li>•Experimentación consiste en la revisión de las características citadas en las plataformas de estudio de forma empírica.</li> <li>•Solicitud de soporte técnico al Equipo de Desarrolladores de Soluciones de Reportes en Seriva S.A.</li> <li>•Revisión de documentos de empresas independientes de evaluación de SW.</li> </ul>	<ul style="list-style-type: none"> <li>•Documentación de los productos,</li> <li>•Ayuda de las herramientas.</li> <li>•Equipo de Desarrolladores de Soluciones de Reportes en Seriva S.A</li> <li>•Referencia Bibliográfica.</li> </ul>
	Capacidad de definición y creación de reportes.	Solución de reportes Administración y despliegue de la solución de reportes Control del archivo del reporte Funcionalidad aprovechable del servidor de reportes Facilidad de uso Capacidad para complementar la solución de reportes con lenguajes o herramientas familiares. Seguridad de la solución de reportes Integración y manejo del visor en la solución de reportes Flexibilidad para establecer la conectividad de la solución de reportes Construcción de reportes Programación de reportes Capacidad para definir reportes variados en una misma solución Flexibilidad para establecer ordenamiento y agrupación sofisticados Definición del formato del reporte Variedad y flexibilidad en la manipulación de objetos del informe Administración y programación referente al paso de parámetros Capacidad para incluir información de la BD Facilidad de navegación entre objetos y repositorios Capacidad para incorporar características propias de la aplicación y el cliente.		

#### **4.1.3. Coeficiente de Correlación de Pearson**

Para la comprobación de la hipótesis que establece la relación directa entre dos variables, en el presente trabajo se ha utilizado el estadístico del coeficiente de correlación de Pearson. Este índice nos indica el grado de relación existente entre dos variables. Puede oscilar entre 1 y -1. Cuando el coeficiente es próximo a 0 es que no existe relación entre las variables.

Con un nivel de significancia  $\alpha = 0.05 = 5\%$  y un número de grados de libertad de  $v = n - L$ , siendo  $n$  el número de características evaluadas y  $L$  el número de variables que intervienen;  $v = 38 - 2 = 36$ . En la tabla de significación del coeficiente de correlación de Pearson basada en la ley de Snodcor (Ver anexo 1). Se tiene que el coeficiente será significativo si es igual o superior a 0.325.

De forma que la hipótesis se expresa como sigue:

$$H_0 : |r_{xy}| < 0.325$$

$$H_1 : |r_{xy}| \geq 0.325$$

$$v = 36$$

$$\alpha = 0.05$$

Siendo:

**X** = Variable independiente: Características de la plataforma para la elaboración de informes.

**Y** = Variable dependiente: Capacidad de definición y creación de reportes.

La siguiente tabla en base a los resultados obtenidos en la sección 2.12, muestra la relación de indicadores de las variables de la hipótesis.

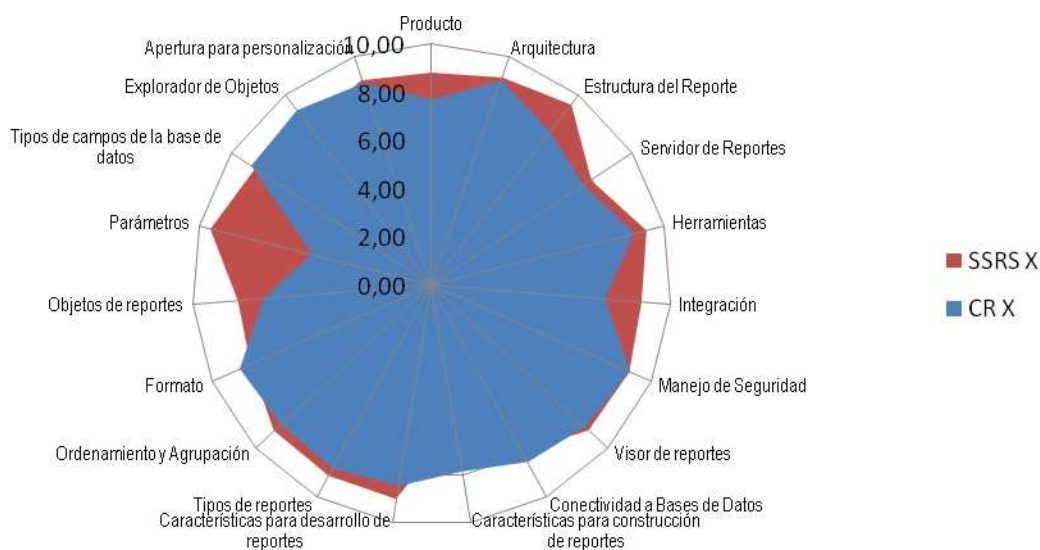
**Tabla IV-3.** Cuadro resumen de comparación CR y SSRS

Indicadores		CR		SSRS	
X	Y	X	Y	X	Y
Producto	Solución de reportes	7.67	7.67	8.83	8.67
Arquitectura	Administración y despliegue de la solución de reportes	8.96	8.67	9.13	9.67
Estructura del Reporte	Control del archivo del reporte	8.00	8.50	9.50	9.75
Servidor de Reportes	Funcionalidad aprovechable del servidor de reportes	7.54	7.71	7.96	9.29
Herramientas	Facilidad de uso	8.67	8.88	9.25	9.25
Integración	Capacidad para complementar la solución de reportes con lenguajes o herramientas familiares.	7.21	7.50	8.79	9.25
Manejo de Seguridad	Seguridad de la solución de reportes	9.00	8.75	9.00	9.00
Visor de reportes	Integración y manejo del visor en la solución de reportes	8.70	9.00	8.90	9.13
Conectividad a Bases de Datos	Flexibilidad para establecer la conectividad de la solución de reportes	8.39	7.80	7.86	7.50
Características para construcción de reportes	Construcción de reportes	7.88	8.33	6.00	5.33
Características para desarrollo de reportes	Programación de reportes	8.50	8.00	9.00	9.50
Tipos de reportes	Capacidad para definir reportes variados en una misma solución	8.71	8.50	9.00	9.00
Ordenamiento y Agrupación	Flexibilidad para establecer ordenamiento y agrupación sofisticados	8.58	8.00	8.92	9.50
Formato	Definición del formato del reporte	8.75	9.50	8.23	9.30
Objetos de reportes	Variedad y flexibilidad en la manipulación de objetos del informe	7.03	8.67	8.08	8.75
Parámetros	Administración y programación referente al paso de parámetros	5.17	7.67	9.50	9.50
Tipos de campos de la base de datos	Capacidad para incluir información de la BD	9.00	9.00	8.80	8.50
Explorador de Objetos	Facilidad de navegación entre objetos y repositorios	9.17	9.50	8.33	8.50
Apertura para personalización	Capacidad para incorporar características propias de la aplicación y cliente en la solución de reportes.	8.83	9.00	9.00	9.50

Las figura **Figura XXXI**. Sobreposición de cacterísticas de la plataforma para la elaboración de informes y la figura **Figura XXXII. Sobreposición capacidad de definición y creación de reportes** muestran una sobreposición tanto de las características de las

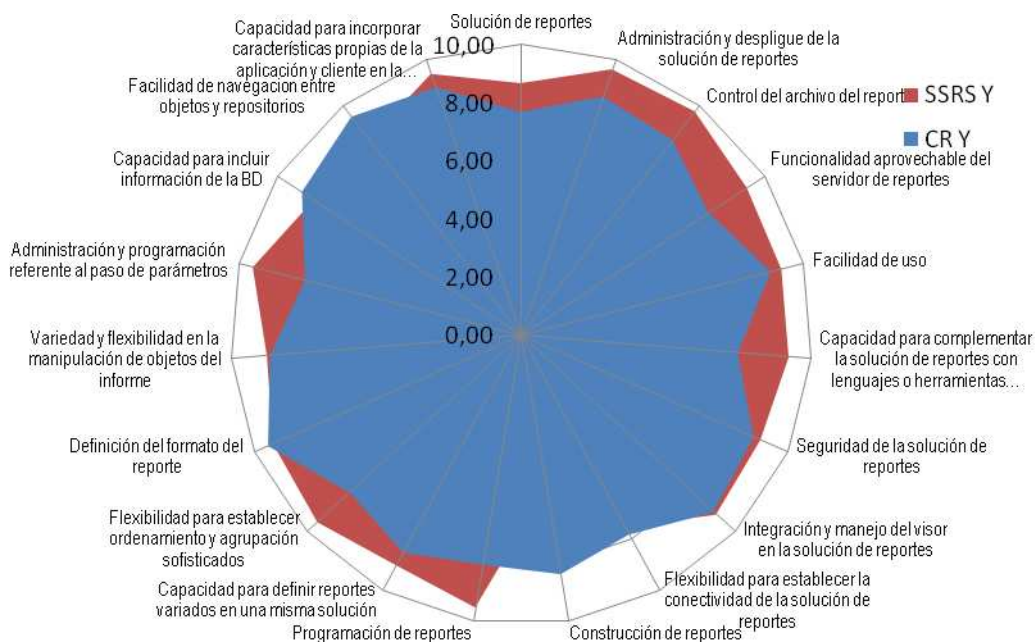


plataformas SAP Business Crystal Reports y Microsoft SQL Server Reporting Services (CR X - SSRS X), así como de la capacidad para la elaboración de informes con cada una de estas herramientas (CR Y - SSRS Y).



**Figura XXXI.** Sobreposición de características de la plataforma para la elaboración de informes

Se puede observar que SSRS abarca una región más amplia que CR, mientras que CR se impone a SSRS en características como conectividad y exploración de objetos. Ninguna herramienta cumple con 100% las características que debería tener una plataforma para elaboración de informes ideal.



**Figura XXXII.** Sobreposición capacidad de definición y creación de reportes

La figura Figura XXXII. Sobreposición capacidad de definición y creación de reportes que indica la capacidad para definición y creación de informes, indica también una sobreposición entre las dos herramientas evaluadas. Al igual que la figura anterior SSRS abarca una región más amplia con respecto a CR.

**Tabla IV-4.** Cuadro de valores para el cálculo del coeficiente de Pearson

n	X	Y	XY	X2	Y2
1	7.6667	7.6667	58.7778	58.7778	58.7778
2	8.9583	8.6667	77.6389	80.2517	75.1111
3	8.0000	8.5000	68.0000	64.0000	72.2500
4	7.5357	7.7143	58.1327	56.7870	59.5102
5	8.6667	8.8750	76.9167	75.1111	78.7656
6	7.2143	7.5000	54.1071	52.0459	56.2500
7	9.0000	8.7500	78.7500	81.0000	76.5625
8	8.7000	9.0000	78.3000	75.6900	81.0000
9	8.3929	7.8000	65.4643	70.4401	60.8400
10	7.8750	8.3333	65.6250	62.0156	69.4444
11	8.5000	8.0000	68.0000	72.2500	64.0000
12	8.7143	8.5000	74.0714	75.9388	72.2500
13	8.5833	8.0000	68.6667	73.6736	64.0000
14	8.7500	9.5000	83.1250	76.5625	90.2500
15	7.0313	8.6667	60.9375	49.4385	75.1111
16	5.1667	7.6667	39.6111	26.6944	58.7778
17	9.0000	9.0000	81.0000	81.0000	81.0000
18	9.1667	9.5000	87.0833	84.0278	90.2500
19	8.8333	9.0000	79.5000	78.0278	81.0000
20	8.8333	8.6667	76.5556	78.0278	75.1111
21	9.1250	9.6667	88.2083	83.2656	93.4444
22	9.5000	9.7500	92.6250	90.2500	95.0625
23	7.9643	9.2857	73.9541	63.4298	86.2245
24	9.2500	9.2500	85.5625	85.5625	85.5625
25	8.7857	9.2500	81.2679	77.1888	85.5625
26	9.0000	9.0000	81.0000	81.0000	81.0000
27	8.9000	9.1667	81.5833	79.2100	84.0278
28	7.8571	7.5000	58.9286	61.7347	56.2500
29	6.0000	5.3333	32.0000	36.0000	28.4444
30	9.0000	9.5000	85.5000	81.0000	90.2500
31	9.0000	9.0000	81.0000	81.0000	81.0000
32	8.9167	9.5000	84.7083	79.5069	90.2500
33	8.2300	9.3000	76.5390	67.7329	86.4900
34	8.0813	8.7500	70.7109	65.3066	76.5625
35	9.5000	9.5000	90.2500	90.2500	90.2500
36	8.8000	8.5000	74.8000	77.4400	72.2500
37	8.3333	8.5000	70.8333	69.4444	72.2500
38	9.0000	9.5000	85.5000	81.0000	90.2500
<b>Σ</b>	<b>319.8318</b>	<b>329.5583</b>	<b>2795.2343</b>	<b>2722.0827</b>	<b>2885.3928</b>

### Gráfica de Dispersión

La gráfica de dispersión permite establecer visualmente patrones que indiquen que las variables están relacionadas. Si esto sucede, se puede apreciar que tipo de línea, o ecuación de estimación, describe la relación.

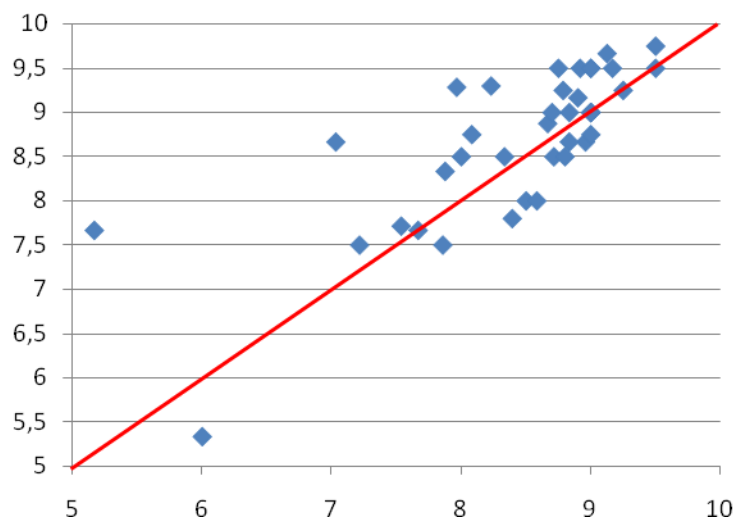


Figura XXXIII. Gráfica de dispersión de las variables de la hipótesis

### Cálculo del coeficiente de Pearson

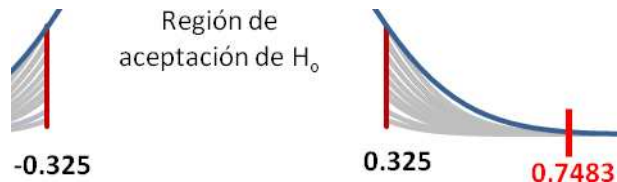
$$r_{xy} = \frac{n \sum XY - \sum X \sum Y}{\sqrt{[n \sum X^2 - (\sum X)^2][n \sum Y^2 - (\sum Y)^2]}}$$

$$r_{xy} = \frac{38(2795,2343) - (319,8318)(329,5583)}{\sqrt{[38(2722,0827) - (319,8318)^2][38(2885,3926) - (329,5583)^2]}}$$

$$r_{xy} = \frac{106218,9034 - 105483,2243}{\sqrt{[108439,1426 - 102292,3803][109644,9264 - 108608,6781]}}$$

$$r_{xy} = \frac{815,6791}{\sqrt{[1146,7623][1036,2583]}} = \frac{815,6791}{\sqrt{1188336,2177}} = \frac{815,6791}{1090,1084} = 0,7483$$

El coeficiente de correlación entre las variables X y Y, es significativo al 1 por mil por ser en valor absoluto superior a  $r(14, 0.001) = 0.742$



**Figura XXXIV.** Distribución normal del coeficiente de Pearson

Las áreas sombreadas representan la zona de rechazo y la parte en blanco, la zona de aceptación de la hipótesis nula. Se ubica el valor del de la tabla 0.325.

Dado que  $r_{xy} = 0.7483$  con un nivel de significación de 0.001, se rechaza la hipótesis nula  $H_0$ . Por tanto se acepta la hipótesis  $H_1$  comprobando que las características de la plataforma utilizada para la elaboración de informes en una aplicación web inciden directamente de forma positiva sobre la capacidad de definición y creación de reportes.

## CONCLUSIONES

- En el mercado existe una gran variedad de plataformas para el desarrollo de informes, no es posible seleccionar una y establecerla como ideal para todo tipo de desarrollo, al contrario, el uso de una u otra herramienta dependerá de las necesidades de la solución de reportes que se requiera implementar.
- A partir de la comprobación de la hipótesis, se concluye que efectivamente las características de la plataforma utilizada para la elaboración de informes en una aplicación Web, inciden directa y positivamente sobre la capacidad tanto del desarrollador como del usuario final para definir y crear reportes.

- La tendencia en herramientas de presentación de informes, es delegar la capacidad de creación de reportes a los usuarios finales, para que puedan diseñar siempre sus propios reportes, sin necesidad de preguntar a programadores o diseñadores, aspectos complicados, de forma que conocen sus datos sin necesidad de entender los conceptos de base de datos.
- Varias herramientas de desarrollo de reportes, han sido evaluadas, de entre ellas, SAP Business Objects, SQL Server Reporting Services y Cognos han sido identificados como los principales contendores. Tienen alto puntaje en términos de consulta, usabilidad y metadatos. Crystal Reports, la solución de SAP Business Objects, proporciona un completo diseño, y formato, desarrollo de reportes, y facilidad de uso, con limitación en asociación de procesos. Cognos tiene ventajas en términos de diseño, formato y metadatos, muestra limitaciones en escalabilidad de la aplicación y desarrollo de reportes. Crystal Reports es una herramienta flexible en el mercado y es ampliamente reconocida y recomendada. Microsoft SQL Server 2008 provee un fuerte motor OLAP y tiene capacidades analíticas avanzadas. Hyperion/Oracle BI proporciona OLAP general, consultas, y capacidades analíticas avanzadas, así como un fuerte manejo de APIs abiertas con BI Beans, revestidos en metadata y usabilidad. Actuate tiene un fuerte lenguaje para escritores profesionales de reportes.
- En lo referente a definición de reportes programáticamente, se ha identificado a SQL Server Reporting Services como la principal herramienta que brinda mayor flexibilidad para éste propósito, sin embargo la decisión de optar por esta herramienta depende en gran medida de la plataforma y arquitecturas que se requiere soportar, y de forma particular el presupuesto de la solución.
- Las plataformas para elaboración de informes evolucionan rápidamente adaptándose a los cada vez más existentes requisitos del mercado.

## RECOMENDACIONES

- Para seleccionar una plataforma para desarrollo de informes, se recomienda realizarlo posterior a las etapas de Planificación y Análisis y teniendo en cuenta el cumplimiento de funcionalidad requerida por la solución de reportes, se puede tomar como referencia las tablas resultantes de este trabajo o realizar un análisis similar en función de la aparición de nuevas herramientas y versiones. En caso no seguir este orden, se corre el riesgo que el entorno de desarrollo de informes afecte el presupuesto y los requerimientos funcionales de la solución.
- En el caso que el entorno de la solución de reportes esté bajo la plataforma Microsoft, se recomienda optar por SQL Server Reporting Services para en función de los costos e integración con herramientas.
- En el caso de que el entorno de la solución este basado en ambientes Java o Sistemas operativos diferentes a Microsoft se recomienda optar por SAP Business Objects o SAS Web Report Studio.
- Se recomienda tener siempre disponible en toda solución de reportes, un módulo de reportes AdHoc, debido a que provoca independencia y satisfacción en los usuarios finales.
- Se sugiere también considerar los indicadores analizados en el cuadro comparativo de características de CR vs SSRS de la sección 2.12, para tener un concepto claro de la funcionalidad de cualquier plataforma de elaboración de informes.

## RESUMEN

El presente trabajo consiste en un análisis comparativo de las principales plataformas para elaboración de informes en aplicaciones Web, a la vez que determina la herramienta con mejores características para la implementación del producto Dynamic Report Generator que forma parte de la Suite MDC Seriva S.A., empresa auspiciante.

Este estudio se realizó para cubrir la necesidad de un marco referencial de selección de plataformas para elaboración de informes.

Las herramientas software utilizadas son las principales plataformas de elaboración de informes existentes: SAP Business Crystal Reports, Cognos 8, Microsoft SQL Reporting Services, etc. Se aplicó el método inductivo y de forma práctica, análisis de documentación y entrevistas, se realizó una cuantificación de características y de capacidades que brindan dichas plataformas. Se propone un cuadro de indicadores principales que una herramienta de definición de informes debe contener, que puede ser aplicado a cualquier herramienta del mercado.

Se aplicó el coeficiente de correlación de Pearson, a partir del cual se obtuvo una correlación positiva de 0.7483 entre las variables: características de la plataforma para la elaboración de informes y capacidad de definición y creación de reportes.

Se confirma que la plataforma que provee las mejores características para la presentación de informes dinámica es Microsoft Reporting Services, en el presente trabajo se utiliza esta tecnología junto con LINQ y C# de Visual Studio 2008 para la realización del caso práctico.

Se recomienda renovar el presente análisis a medida que nuevas versiones o productos aparezcan en el mercado, de forma que se mantenga continuidad.



## **SUMMARY**

This work is a comparative analysis of the main platforms for reporting in web applications, while determining the best tool with capabilities to implement the Dynamic Report Generator product as part of the MDC Suite Seriva SA, work sponsor.

This study was undertaken to cover the need to establish a framework for selecting platforms for reporting.

The software tools used are essentially all major platforms for reporting on the market, as SAP Crystal Reports, Cognos 8, Microsoft SQL Reporting Services, etc. Applying the inductive method of practical way, analysis of documentation and interviews, this study makes a quantification of the characteristics and capabilities that provide the platforms for the definition and reporting. It proposes a table of key indicators that a tool for defining reports should contain and that can be applied to any reporting tool on the market.

It applied the Pearson correlation coefficient, from which there was a positive correlation of 0.7483 between the variables: characteristics of the platform for reporting and ability to define and create reports.

This work confirmed that the platform that provides the best features for reporting dynamic is Microsoft Reporting Services, for the case study of this work SSRS is used along with LINQ and C # Visual Studio 2008.

It is advisable to renew the present analysis as new versions or products on the market.

## GLOSARIO

**Ad hoc.**- Es una locución latina que significa literalmente «para esto». Generalmente se refiere a una solución elaborada específicamente para un problema o fin preciso y, por tanto, no es generalizable ni utilizable para otros propósitos. Se usa pues para referirse a algo que es adecuado sólo para un determinado fin. En sentido amplio, ad hoc puede traducirse como «específico» o «específicamente». En informática esto implica que el sistema permite al usuario personalizar una consulta en tiempo real, en vez de estar atado a las consultas prediseñadas para informes. Generalmente las consultas ad hoc permiten a los usuarios con poca experiencia en SQL tener el mismo acceso a la información de la base de datos, para esto los sistemas que soportan ad hoc poseen GUIs para generarlas.<sup>1</sup>

**Report prompts.**- Un prompt es una propiedad de un informe que se puede configurar cuando se crea o modifica un reporte. Cuando un usuario ejecuta un reporte, un prompt (pregunta) solicita al usuario que ingrese un valor para un parámetro requerido antes de la presentación del informe.

**WYSIWYG.**- What You See Is What You Get ("lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.<sup>2</sup>

**BI.**- Conjunto de metodologías, procesos, arquitectura y tecnologías que transforman datos primitivos en información significativa y útil utilizada para hacer posible estrategias más efectivas, tácticas, entendimiento operacional, y toma de decisiones.<sup>3</sup>

---

<sup>1</sup> Tomado de Wikipedia. Bajo licencia de documentación libre de GNU, Copy Left .

<sup>2</sup> Tomado de Wikipedia. Bajo licencia de documentación libre de GNU, Copy Left .

<sup>3</sup> Forrester (Evelson, 2008)

## ANEXOS

### Anexo 1. Significación del coeficiente de correlación de Pearson

Tabla 00-1. Significación del coeficiente de correlación de Pearson

Grados de libertad  v	Niveles de significación				
	0.1	0.05	0.02	0.01	0.001
1	0.988	0.997	0.999	1	1
2	0.9	0.95	0.98	0.99	0.999
3	0.805	0.878	0.934	0.959	0.991
4	0.729	0.811	0.882	0.917	0.974
5	0.669	0.754	0.833	0.874	0.951
6	0.622	0.707	0.789	0.834	0.925
7	0.582	0.666	0.75	0.798	0.898
8	0.549	0.632	0.716	0.765	0.872
9	0.521	0.602	0.685	0.735	0.847
10	0.497	0.576	0.658	0.708	0.823
11	0.476	0.553	0.634	0.684	0.801
12	0.458	0.532	0.612	0.661	0.78
13	0.441	0.514	0.592	0.641	0.76
14	0.426	0.497	0.574	0.623	0.742
15	0.412	0.482	0.558	0.606	0.725
16	0.4	0.468	0.542	0.59	0.708
17	0.389	0.456	0.528	0.575	0.693
18	0.378	0.444	0.516	0.561	0.679
19	0.369	0.433	0.503	0.549	0.665
20	0.36	0.423	0.492	0.537	0.652
25	0.323	0.381	0.445	0.487	0.597
30	0.296	0.349	0.409	0.449	0.554
35	0.275	0.325	0.381	0.418	0.519
40	0.257	0.304	0.358	0.393	0.49
45	0.243	0.288	0.338	0.372	0.465
50	0.231	0.273	0.322	0.354	0.443
60	0.211	0.25	0.295	0.325	0.408
70	0.195	0.232	0.274	0.302	0.38
80	0.183	0.217	0.256	0.283	0.357
90	0.173	0.205	0.242	0.267	0.337
100	0.164	0.195	0.23	0.254	0.321

## Anexo 2. Esquema del archivo rdl generado por la herramienta DRG

```

<?xml version="1.0" encoding="utf-8"?>
<Report xmlns:rd="http://schemas.microsoft.com/SQLServer/reporting/reportdesigner"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2008/01/reportdefinition">

  <Body>
    <Height>2in</Height>
    <Style />
    <ReportItems>
      <Tablix Name="Tablix1">
        <DataSetName>DataSet1</DataSetName>
        <TablixBody>
          <TablixColumns>
            <TablixColumn>
              <Width>1.7in</Width>
            </TablixColumn>
          </TablixColumns>
          <TablixRows>
            <TablixRow>
              <Height>0.25in</Height>
              <TablixCells>
                <TablixCell>
                  <CellContents>
                    <Textbox Name="cabvar1">
                      <CanGrow>true</CanGrow>
                      <KeepTogether>true</KeepTogether>
                      <Paragraphs>
                        <Paragraph>
                          <TextRuns>
                            <TextRun>
                              <Value>var 1</Value>
                              <Style>
                                <Color>White</Color>
                                <FontWeight>Bold</FontWeight>
                              </Style>
                            </TextRun>
                          </TextRuns>
                          <Style>
                            <TextAlign>Center</TextAlign>
                          </Style>
                          <rd:DefaultName>cabvar1</rd:DefaultName>
                        </Paragraph>
                      </Paragraphs>
                    </Style>
                    <Border>
                      <Style>Solid</Style>
                    </Border>
                  </CellContents>
                </TablixCell>
              </TablixCells>
            </TablixRow>
          </TablixRows>
          <TablixRow>
            <Height>0.25in</Height>
            <TablixCells>
              <TablixCell>
                <CellContents>
                  <Textbox Name="valorvar1">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                      <Paragraph>
                        <TextRuns>
                          <TextRun>
                            <Value>=Fields!var1.Value</Value>
                          </TextRun>
                        </TextRuns>
                      </Paragraph>
                    </Paragraphs>
                  </CellContents>
                </TablixCell>
              </TablixCells>
            </TablixRow>
          </TablixRows>
          <rd:DefaultName>valorvar1</rd:DefaultName>
        </TablixBody>
        <TablixColumnHierarchy>
          <TablixMembers>
            <TablixMember />
          </TablixMembers>
        </TablixColumnHierarchy>
        <TablixRowHierarchy>
          <TablixMembers>
            <TablixMember>
              <KeepWithGroup>After</KeepWithGroup>
            </TablixMember>
            <TablixMember>
              <Group Name="Details" />
            </TablixMember>
          </TablixMembers>
        </TablixRowHierarchy>
        <Height>0.5in</Height>
        <Width>1in</Width>
        <Style>
          <Border>
            <Style>Solid</Style>
          </Border>
        </Style>
        <Filters>
          <Filter>
            <FilterExpression>=Fields!var1.Value</FilterExpresio
n>
              <Operator>Equal</Operator>
              <FilterValues>
                <FilterValue>
                  "=Parameters!var1.Value"
                </FilterValue>
              </FilterValues>
            </Filter>
          </Filters>
        </Tablix>
      </ReportItems>
    </Body>
    <Width>6.5in</Width>
    <Page>
      <PageHeader>
        <Height>1in</Height>
        <PrintOnFirstPage>true</PrintOnFirstPage>
        <PrintOnLastPage>true</PrintOnLastPage>
      </PageHeader>
      <ReportItems>
        <Textbox Name="TxtTitulo">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
        </PageHeader>
      </ReportItems>
    </Page>
  </Body>
</Report>

```

```

<TextRuns>
<TextRun>
  <Value></Value>
  <Style>
    <FontSize>14pt</FontSize>
    <FontWeight>Bold</FontWeight>
    <Color>Teal</Color>
  </Style>
</TextRun>
</TextRuns>
<Style>
  <TextAlign>Center</TextAlign>
</Style>
</Paragraph>
</Paragraphs>
<rd:DefaultName>Textbox1</rd:DefaultName>
<Top>0.21542in</Top>
<Height>0.25in</Height>
<Width>6.5in</Width>
<Style>
<Border>
  <Style>None</Style>
</Border>
<PaddingLeft>2pt</PaddingLeft>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
<PaddingBottom>2pt</PaddingBottom>
</Style>
</Textbox>
</ReportItems>
<Style>
<Border>
  <Style>None</Style>
</Border>
</Style>
</PageHeader>
<LeftMargin>1in</LeftMargin>
<RightMargin>1in</RightMargin>
<TopMargin>1in</TopMargin>
<BottomMargin>1in</BottomMargin>
<Style />
</Page>
<rd:ReportID>a542eaea-7fd4-431d-87ac-7bcd58eaf897</rd:ReportID>
<rd:ReportUnitType>Inch</rd:ReportUnitType>
<DataSources>
  <DataSource Name="FuenteDatos">
<DataSourceReference>UBICACION</DataSourceReference>
  <rd:DataSourceID>9153961b-13fb-47bd-b8ec-b25013eec38d</rd:DataSourceID>
</DataSource>
</DataSources>
<ReportParameters>
  <ReportParameter Name="var 1">
    <DataType>String</DataType>
    <Prompt>var1</Prompt>
    <ValidValues>
      <DataSetReference>
        <DataSetName>DataSet1</DataSetName>
        <ValueField>var 1</ValueField>
        <LabelField>var 1</LabelField>
      </DataSetReference>
    </ValidValues>
  </ReportParameter>
</ReportParameters>
<DataSets>
  <DataSet Name="DataSet1">
    <Query>
<DataSourceName>FuenteDatos</DataSourceName>

```

```

<CommandText>
  select
    [MERCY_PC_DBSERVERUBICACIONdboPROVINCIA].NOMBRE as var1
  from
    [MERCY-PC\DBSERVER].[UBICACION].[dbo].[PAIS] as [MERCY_PC_DBSERVERUBICACIONdboPAIS]
  INNER JOIN [MERCY-PC\DBSERVER].[UBICACION].[dbo].[PERSONA] AS
[MERCY_PC_DBSERVERUBICACIONdboPERSONA]
  ON [MERCY_PC_DBSERVERUBICACIONdboPAIS].CODPAIS
[MERCY_PC_DBSERVERUBICACIONdboPERSONA].CODPAIS
  INNER JOIN [MERCY-PC\DBSERVER].[UBICACION].[dbo].[ALCALDE] AS
[MERCY_PC_DBSERVERUBICACIONdboALCALDE]
  ON [MERCY_PC_DBSERVERUBICACIONdboALCALDE].CODALCALDE
[MERCY_PC_DBSERVERUBICACIONdboALCALDE].CODALCALDE
  INNER JOIN [MERCY-PC\DBSERVER].[UBICACION].[dbo].[CANTON] AS
[MERCY_PC_DBSERVERUBICACIONdboCANTON]
  ON [MERCY_PC_DBSERVERUBICACIONdboALCALDE].CODCANTON
[MERCY_PC_DBSERVERUBICACIONdboCANTON].CODCANTON
  INNER JOIN [MERCY-PC\DBSERVER].[UBICACION].[dbo].[PROVINCIA] AS
[MERCY_PC_DBSERVERUBICACIONdboPROVINCIA]
  ON [MERCY_PC_DBSERVERUBICACIONdboCANTON].CODPROVINCIA
[MERCY_PC_DBSERVERUBICACIONdboPROVINCIA].CODPROVINCIA
</CommandText>
<rd:UseGenericDesigner>true</rd:UseGenericDesigner>
</Query>
<Fields>
  <Field Name="var1">
    <DataField>var1</DataField>
    <rd:TypeName>System.String</rd:TypeName>
  </Field>
</Fields>
</DataSet>
</DataSets>
</Report>

```

# BIBLIOGRAFÍA

## LIBROS

1. LACHEV, T. Microsoft Reporting Services in Action. Atlanta, GA: Manning Publications Co, 2004. 656 p.
2. MARTÍNEZ, L.; MOSQUERA, G. Hipótesis Estadística con Aplicaciones: Versión Electrónica. 3<sup>ra</sup> ed. Caracas: ISID., 2004. 294 p.
3. PIALORSI, P.; RUSSO, M. Introducing Microsoft LINQ. Microsoft Press. Washington: Microsoft Press, 2007. 203 p.
4. PRESSMAN, R. Ingeniería de Software. Un enfoque práctico. 5<sup>ta</sup> ed. Barcelona: McGraw-Hill, 2002. 640 p.
5. SZALL, K. Analyzing Requirements and Defining Microsoft .NET Solution Architectures. Washington: Microsoft PSSRS, 2003. 524 p.
6. TROELSEN, A. Pro C# 2008 and the .NET 3.5 Platform. 4<sup>ta</sup> ed. New York: Apress, 2007. pp 69 – 182, 415 – 469.

## RECURSOS DE INTERNET

1. ACTUATE. Release Notes Actuate 9 Service Pack 2. 20 de Julio de 2007. (Documento PDF). [www.xlsoft.com/jp/products/actuate/pdf/actuate9sp2\\_releasenote.pdf](http://www.xlsoft.com/jp/products/actuate/pdf/actuate9sp2_releasenote.pdf)  
2008 07 25.
2. BLANCO, S. Comparación entre Active Reports, Crystal Reports, y MS Reporting Services. 25 de Julio de 2005. (Documento PDF).  
[www.emagister.com/uploads\\_user\\_home/Comunidad\\_Emagister\\_2462\\_Comp\\_aracion\\_Reports.pdf](http://www.emagister.com/uploads_user_home/Comunidad_Emagister_2462_Comp_aracion_Reports.pdf)  
2008 01 15.
3. DÍAZ, F., & SÁNCHEZ, C. Evento SSRS2005 vs Crystal. SNUG. 28 de Marzo de 2008. (Documento PDF).  
[http://www.crystalreportsbook.com/forum/forum\\_posts.asp?TID=1529](http://www.crystalreportsbook.com/forum/forum_posts.asp?TID=1529)  
2008 04 20.
4. EQUIPO DANYSOFT. ¿Crystal Reports o Reporting Services?. 2006. (Documento Word doc). [www.crystalreportsbook.com/SSRSandCR\\_ExecSummary.asp](http://www.crystalreportsbook.com/SSRSandCR_ExecSummary.asp)  
2008 06 18.

5. EVELSON, B. The Forrester Wave™: Enterprise Business Intelligence Platforms, Q3 2008 for Information & Knowledge Management Professionals. 31 de Julio de 2008. (Documento PDF).  
[www.sas.com/news/analysts/forresterwave\\_bi\\_0708.pdf](http://www.sas.com/news/analysts/forresterwave_bi_0708.pdf)  
2008 05 17.
6. GILE, K. The Forrester Wave™: BI Reporting And Analysis Platforms, Q1 2006. 8 de February de 2006. (Documento PDF).  
[www.cognos.com/news/forrester\\_wave\\_bi.pdf](http://www.cognos.com/news/forrester_wave_bi.pdf)  
2008 04 02.
7. GLOBAL KNOWLEDGE NETWORK. SQL Server 2005 Reporting Services vs. Crystal Reports XI Developer Edition. 2005. (Documento PDF).  
[http://images.globalknowledge.com/wwwimages/whitepaperpdf/SQLvsCrystal\\_WP.pdf](http://images.globalknowledge.com/wwwimages/whitepaperpdf/SQLvsCrystal_WP.pdf)  
2008 03 30.
8. GONZÁLEZ, J. SQL Server 2008: Creando informes con SSRS 2008. 8 de Mayo de 2008. (Documento PDF).  
<http://geeks.ms/blogs/ciin/archive/2008/05/08/sql-server-2008-creando-informes-con-ssrs-2008-ii.aspx>  
2008 04 20.
9. MICROSTRATEGY. MicroStrategy 8 Advanced Reporting Guide. Julio de 2005. (Documento PDF).  
<http://polymorphe.free.fr/cours/bi/microstrategy/man-qra-000-adr-0-ver801-001.pdf>  
2009 01 16.
10. NUCLEUS RESEARCH INC. Microsoft Office PerformancePoint Server 2007. Boston. 2008. (Documento PDF).  
<http://nucleusresearch.com/library/microsoft-roi/i116.pdf>  
2008 10 15.
11. ORACLE BUSINESS INTELLIGENCE SUITE ENTERPRISE EDITION PLUS – Technical Overview. Noviembre de 2007. (Documento PDF).  
[www.oracle.com/appserver/business-intelligence/enterprise-edition.html](http://www.oracle.com/appserver/business-intelligence/enterprise-edition.html)  
2009 01 16.

12. ORACLE. New Features in oracle Business Intelligence Suite Enterprise Edition 10g Release 3. Enero de 2007. (Documento PDF).  
[www.oracle.com/appserver/business-intelligence/docs/oracle-bi-ee-new-features-10gr3.pdf](http://www.oracle.com/appserver/business-intelligence/docs/oracle-bi-ee-new-features-10gr3.pdf)  
2009 01 16.
13. PATRA, S. A Comparative Analysis of SQL Server Reporting Services and Crystal Reports. Junio de 2008. (Documento PDF).  
[www.silicusblogs.com/kb/.../a-comparative-analysis-of-sql-server-reporting-services-and-crystal-reports.html](http://www.silicusblogs.com/kb/.../a-comparative-analysis-of-sql-server-reporting-services-and-crystal-reports.html)  
2008 04 20.
14. SNEDECOR, I. Oracle Reports Building Reports, 10g Release 2 (10.1.2). Julio de 2005. (Documento PDF).  
[www.oracle.com/technology/documentation/reports.html](http://www.oracle.com/technology/documentation/reports.html)  
2009 01 16.
15. STINAFF, E. Hyperion System 9 BI+ Financial Reporting: Tips and Tricks for Designing Advanced Reports. 1 de Marzo de 2008. (Documento PDF).  
[www.oaug.org/conferencesandeducation/conferences/2008/collaborate08/agenda/hyperionsessions.pdf](http://www.oaug.org/conferencesandeducation/conferences/2008/collaborate08/agenda/hyperionsessions.pdf)  
2009 01 16.
16. WELCKER, B. (s.f.). What's New in SQL Server 2008. (Documento PDF).  
[www.microsoft.com/sqlserver/2008/en/us/overview.aspx](http://www.microsoft.com/sqlserver/2008/en/us/overview.aspx)  
2008 07 15.
17. PRESLAN, L., NEWMARK, E., & HAGERTY, J. Customer Analytics: 85 Percent Do Not Achieve ROI—Know Why Before You Invest. AMR Reseach. 2004.(Documento PDF).  
2008 04 28.
18. SCHLEGEL, K., & THEISSEN, M. Enterprise Reporting and Dashboard Assessment: Part 2 Vendor Scoring Summary. META Group, Inc. 2005. (Documento PDF).  
2008 04 24.
19. TOMSON, B. SQL Server 2005 Reporting Services vs. Crystal Reports XI Developer Edition. Global Knowledge Network. 2005. (Documento PDF).  
2008 04 18.



## **SITIOS WEB**

### **DESARROLLO DE REPORTE**

1. <http://www.enterprise-dashboard.com/2006/11/30/dashboard-spy-reader-compares-enterprise-reporting-products/>
2. <http://www.microsoft.com/sql/technologies/reporting/default.mspx>
3. <http://msdn.microsoft.com/en-us/library/bb630404.aspx>
4. <http://www.microsoft.com/msdn>
5. <http://www.desarrolloweb.com/articulos/>
6. <http://www.microsoft.com/sql/reporting/default.asp>

### **BUSINESS OBJECTS**

7. [http://support.businessobjects.com/communityCS/TechnicalPapers/cr10\\_UserGde\\_es.zip](http://support.businessobjects.com/communityCS/TechnicalPapers/cr10_UserGde_es.zip)
8. [http://support.businessobjects.com/communityCS/TechnicalPapers/ce10\\_user\\_en.pdf](http://support.businessobjects.com/communityCS/TechnicalPapers/ce10_user_en.pdf)
9. [http://support.businessobjects.com/communityCS/TechnicalPapers/ce9\\_viewers.zip](http://support.businessobjects.com/communityCS/TechnicalPapers/ce9_viewers.zip)
10. <http://www.bcp.com.pe>
11. <http://www.microsoft.com/sql/2005>
12. <http://www.crystaldecisions.com>

### **MICROSOFT SQL SERVER 2008**

13. <http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>
14. <http://msdn.microsoft.com/en-us/sqlserver/bb671149.aspx>
15. <http://www.microsoft.com/sql/>
16. <http://msdn2.microsoft.com/sqlserver>
17. <http://technet.microsoft.com/sqlserver>
18. <http://msdn2.microsoft.com/sqlserver>
19. <http://technet.microsoft.com/sqlserver>
20. <http://www.microsoft.com/sql/reporting>
21. <http://microsoft.com/msdn>
22. <http://microsoft.com/technet>

### **LABORATORIOS VIRTUALES**

23. <http://www.microsoft.com/technet/traincert/virtuallab/rms.mspx>

### **GRUPOS DE NOTICIAS**

24. <http://communities2.microsoft.com/communities/newsgroups/en-us/default.aspx>

### **SITIOS DE COMUNIDADES TÉCNICAS**

25. <http://www.microsoft.com/communities/default.mspx>

## **GRUPOS DE USUARIOS**

26. <http://www.microsoft.com/communities/usergroups/default.aspx>

## **VARIOS**

27. <http://www.ssw.com.au/ssw/Company/EnterpriseReporting.aspx#getstarted#getstarted>

28. <http://www.builderau.com.au/program/0,39024614,39173725-6,00.htm>

29. [http://www.avdf.com/june98/art\\_r002.html](http://www.avdf.com/june98/art_r002.html)

30. [http://www.taltech.com/TALtech\\_web/support/bcax/Crystal\\_Reports.htm](http://www.taltech.com/TALtech_web/support/bcax/Crystal_Reports.htm)

31. [http://www.themsteam.com/Offerings/alternative\\_crystal\\_report\\_development.aspx](http://www.themsteam.com/Offerings/alternative_crystal_report_development.aspx)

32. <http://www.eggheadcafe.com/articles/20040619.asp>

33. <http://www.xtras.net/Products/activerereports.asp>

34. <http://en.wikipedia.org>

35. <http://es.wikipedia.org>