

Servidor Dinámico de Reportes

Keimer Montes Oliver

Universidad de las Ciencias Informáticas, La Habana, Cuba, kmontes@uci.cu

Beatriz Hernández Cervantes

Universidad de las Ciencias Informáticas, La Habana, Cuba, bcervante@uci.cu

Yoandry Pacheco Aguila

Universidad de las Ciencias Informáticas, La Habana, Cuba, andypa@uci.cu

ABSTRACT

At present, the reports generation from data stored in a specific data source has become a pillar of the institutions decision-making. Considering how quickly a report is generated largely depends on the engine used to compile, fill data and export the report in multiple formats, this research aims to develop a high performance report engine for the Report Dynamic Generator. The result was the Reports Dynamic Server (SDR), developed entirely with free technologies; it is a separate system based on a service-oriented architecture, which can be easily integrated with applications responsible for design of the reports regardless of the programming language in which they were developed. Guarantees the management and export reports and subreports to multiple formats, also allows scheduling automatic tasks, the statistical calculation of the use of the reports, and provides a powerful user interface for the visual management of server resources. In the present article besides discussed the results of the tests for checking the quality of the developed system.

Keywords: Reports Dynamic Server (SDR), report, subreport, JasperReport.

RESUMEN

En la actualidad, la generación de reportes a partir de la información almacenada en una determinada fuente de datos se ha convertido en un pilar de la toma de decisiones de las entidades. Teniendo en cuenta que la rapidez con se genera un reporte depende en gran medida del motor que se utiliza para compilar, llenar los datos y exportar el reporte en múltiples formatos, la presente investigación tiene como objetivo desarrollar un motor de reportes de altas prestaciones para el Generador Dinámico de Reportes. Se obtuvo como resultado el Servidor Dinámico de Reportes, desarrollado completamente con tecnologías libres, el cual constituye un sistema independiente basado en una arquitectura orientada a servicios, que puede integrarse fácilmente a las aplicaciones encargadas de diseñar los reportes independientemente del lenguaje de programación en que hayan sido desarrolladas. Garantiza la gestión y exportación de los reportes y subreportes a múltiples formatos, además permite la programación de tareas automáticas, el cálculo estadístico del uso de los reportes, y brinda una potente interfaz de usuario para la administración visual de los recursos del servidor. En el presente artículo también se exponen los resultados de las pruebas realizadas para la comprobación de la calidad del sistema desarrollado.

Palabras claves: Servidor Dinámico de Reportes (SDR), reporte, subreporte, JasperReport.

1. INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TICs) son una rama del saber que caracteriza la sociedad actual, donde la información juega un papel determinante en el desarrollo económico y es un factor clave para la búsqueda de nuevas alternativas que ayuden al proceso de toma de decisiones, la administración y el control estadístico de cualquier entidad moderna (Chen, 2008). La acumulación de la información es un comportamiento común en la mayoría de las empresas, la creación de reportes de forma dinámica a partir de la información almacenada contribuye notablemente a la toma de decisiones al disminuir el tiempo y esfuerzo en el análisis de los datos (Calzada, 2009).

Cornella afirma que “las TIC constituyen un verdadero rol catalítico, capacitador en la transformación de la sociedad industrial en la sociedad de la información”, y que “el verdadero objetivo de las TIC debe ser el aprovechamiento estratégico de la información para la toma de decisiones” (Cornella, 1994) por lo que éstas deben ser incorporadas a las instituciones, para lograr mayor eficiencia en el ámbito institucional.

La Universidad de las Ciencias Informáticas (UCI), forma parte de las nuevas alternativas que se han llevado a cabo en Cuba para lograr el avance vertiginoso en las TICs. En su estructura productiva cuenta con varios centros de desarrollo de software entre los que se encuentra el Centro de Tecnologías de Gestión de Datos (DATEC), con la misión de crear bienes y servicios informáticos relacionados con la gestión de datos, área del conocimiento que agrupa tanto a los sistemas de información, como a los denominados sistemas de inteligencia empresarial o de negocios, cuyo propósito fundamental es apoyar el proceso de toma de decisiones. Este centro cuenta con prestigio nacional e internacional, reconocido por la integración de soluciones, productos y servicios que se encargan de la captura, análisis y visualización de la información almacenada en múltiples fuentes de datos.

El Generador Dinámico de Reportes (GDR) es uno de los sistemas informáticos desarrollados por DATEC, que ha sido desplegado en múltiples instituciones tanto nacionales como internacionales con gran éxito y aceptación; pero la velocidad y dinamismo con que evolucionan las TICs, el constante perfeccionamiento que caracteriza la producción de herramientas informáticas, y la necesidad de garantizar la generación de reportes más complejos, confiables, y de forma cada vez más rápida a pesar del creciente cúmulo de información, determinó la actualización del sistema informático utilizando nuevas tecnologías o nuevas versiones de tecnologías ya empleadas, con el consecuente cambio en la arquitectura del GDR.

Por lo que, teniendo en cuenta que la rapidez con que se genera un reporte depende en gran medida del motor en el cual se apoya para compilar, llenar los datos y exportar el reporte a un formato específico, constituye un hito en la definición de la arquitectura del GDR 2.0 la selección del motor de reportes; sobretodo valorando que la versión anterior empleaba PHPReport, pero ya que la nueva versión implementará nuevos requisitos de diseño de reportes que no son soportados por ese motor, como es el caso de los subreportes, y además de que el mencionado motor ya no cuenta con soporte técnico, fue evidente la necesidad del cambio de tecnología del motor de reportes.

2. MATERIALES Y MÉTODOS

Fue necesario guiar la investigación utilizando métodos científicos, tanto teóricos como empíricos. Los métodos teóricos permitieron estudiar las características del objeto de la investigación. Dentro de estos se empleó, el analítico – sintético, para buscar la esencia de los fenómenos y caracterizar los elementos más importantes relacionados con los motores de reportes más utilizados y con mayores prestaciones. Además ayudó en el establecimiento de los métodos, herramientas y procedimientos más factibles para la implementación del servidor de reportes que se propone. También se empleó el método inductivo – deductivo para generalizar el conocimiento adquirido luego de aceptada la idea a defender y determinar que el desarrollo de un servidor para la generación de reportes es una idea bien aceptada. Otro método utilizado es el histórico – lógico con el objetivo de actualizar los antecedentes y el estado actual del desarrollo de motores de reportes, así como los logros y limitaciones de los mismos en Cuba y en el mundo.

3. DESARROLLO

Una vez confirmadas todas las limitaciones que impedían que la nueva versión del GDR utilizara PHPReport como motor de reportes, fue concluyente la necesidad de realizar el cambio tecnológico de esa parte de la arquitectura para la versión 2.0, por lo que se construyó el marco teórico sobre los motores de reportes más utilizados y con mejores prestaciones, con el objetivo de establecer una comparación que sirviera de referencia en la selección del nuevo motor de reportes. A continuación se expone un breve resumen del estudio realizado.

Los motores de reportes son principalmente los encargados de obtener el diseño del reporte en formato .jrxml para realizar el proceso de compilado del fichero, llenado de los datos a partir de la información guardada en la fuente de datos y luego exportarlo en un determinado formato.

Crystal Reports

Crystal Reports facilita la creación de reportes simples y dispone también de poderosas herramientas necesarias para generar reportes complejos o especializados. Está diseñado para generar el reporte prácticamente desde

cualquier origen de datos. Los asistentes incorporados guían al usuario paso a paso a través de la creación de reportes y la ejecución de tareas comunes relacionadas con el uso de reportes. Los gráficos proporcionan información en forma visual cuando las palabras y números no son suficientes. (SAP, 2013)

Características y funcionalidades principales:

- Puede ser utilizado para interactuar con el reporte mediante programación en tiempo de ejecución, usando uno de los cuatro modelos de objetos posibles.
- La presentación de reportes, se realiza tanto en aplicaciones escritorio como web, de manera interactiva y proporciona funciones como la profundización en gráficos, la exploración de informes y la búsqueda de texto.
- Brinda opciones de salida flexibles incluyendo los populares formatos PDF, HTML (HyperText Markup Language), Excel, entre otros.

Jaspers Server

Jaspers Server es un software propietario bajo licencia Dual¹, está compuesto por un conjunto de librerías Java² para facilitar la generación de reportes en aplicaciones web y de escritorio. Está escrito completamente en Java y es capaz de utilizar los datos procedentes de cualquier tipo de fuente de datos y presentar los documentos con precisión de píxel, dichos documentos se pueden ver, imprimir o exportar en una gran variedad de formatos. (NAN -TIC, 2013)

Características y funcionalidades principales:

- Contiene Scriptlets, que pueden acompañar a la definición del informe y pueden ser invocados en cualquier momento, como un procesamiento adicional. Los scriptlets son fragmentos de código Java embebido en HTML, que se pueden invocar en el proceso de generación de informes.
- Permite imprimir o exportar en una variedad de formatos de documentos incluyendo HTML, PDF, Excel, Open Office y Word.
- Permite la generación de subreportes.

A modo de conclusión, la siguiente tabla muestra el resultado del estudio a través de los criterios comparativos trascendentales para la investigación:

Tabla 1. Comparación entre los motores de reportes estudiados.

Propiedades	Crystal Reports	Jasper Server
Sistema Operativo	Windows	Multiplataforma
Fuentes de datos	Múltiples	Múltiples
Lenguaje de Programación	.Net	Java
Servicios web	Si	Si
Subreportes	Si	Si
Formatos de exportación	PDF, XML, HTML, CSV, XLS, RTF, TXT, JPEG, DOC, ODT, PPTX, DOCX, ODS, XSIX	PDF, XML, HTML, CSV, XLS, RTF, TXT, JPEG, DOC, ODT, PPTX, DOCX, ODS, XSIX
Licencia	Privativa	Dual

Donde puede comprobarse que a pesar de las excelentes características de los servidores analizados tienen como principal desventaja la licencia por la cual se rigen, pues Crystal Reports es un software privativo y su uso acarrearía un costo adicional para las herramientas que lo utilicen y el Jaspers Server se puede utilizar con fines académicos pero no con fines comerciales, por lo que no contribuyen a la libertad tecnológica aspirada y debilita las estrategias establecidas en ese sentido.

¹Licencia Dual: Se puede utilizar con fines académico, pero no se puede utilizar para la comercialización.

² Java: Lenguaje de programación orientado a objetos.

Partiendo de estos resultados, la decisión tomada fue desarrollar un motor de reportes que presentara las principales características de los servidores anteriormente mencionados y se le incorporan otras en función de los intereses del GDR. De ese modo surgió el Servidor Dinámico de Reportes (SDR), que utilizando las librerías de código abierto de JasperReport 5.2, aprovecha sus potencialidades y provee la solución al problema existente.

3.1 METODOLOGÍA, HERRAMIENTAS Y TECNOLOGÍAS

Antes de iniciar el desarrollo del servidor, se realizó una profunda revisión bibliográfica que constituyó la base para la correcta selección de la metodología de desarrollo del software que guiaría el trabajo, además de las herramientas y tecnologías necesarias para la implementación del SDR. Como resultado se utilizó el lenguaje de programación Java, específicamente la plataforma J2EE (Java 2 Enterprise Edition); teniendo en cuenta que una especificación precisa era la necesidad de que funcionara como un servidor que publicara servicios que serían consumidos a través de la red a través el servidor Apache Tomcat, se seleccionó la arquitectura REST (Representational State Transfer) por lo que constituye un sistema con arquitectura orientada a servicios. Se escogió el Entorno Integrado de Desarrollo (IDE por las siglas en inglés de Integrated Developed Environment) Netbeans en su versión 7.4, por ser una herramienta de software libre. Como gestor de base datos PostgreSQL en su versión 9.1 para almacenar la información que se administra en el servidor. También se eligió la metodología ágil OpenUP como guía para el desarrollo, apoyada en a la herramienta CASE³ (Cumputer Aided Software Engineering) Visual Paradigm en su versión 8.0 para el modelado del sistema con el lenguaje de modelado UML (Unified Modeling Language), que permite representar gráficamente el ciclo de vida completo del desarrollo de software.

3.2 SERVIDOR DINÁMICO DE REPORTES, PRINCIPALES CARACTERÍSTICAS.

Un servidor de reportes tiene la misión de proveer a los usuarios una manera transparente y rápida de exportar informes, utilizando reportes previamente diseñados y parametrizados, además de garantizar la seguridad de los procesos.

El SDR está basado en una arquitectura orientada a servicios utilizando servicios de Transferencia de Estado Representacional (REST por las siglas en inglés de Representational State Transfer), esta arquitectura fue ganando amplia aceptación en toda la web como una alternativa más simple a SOAP⁴(Simple Object Access Protocol), ya que permite utilizar los servicios sin estado; plantea que el servidor es responsable de generar las respuestas y proveer una interfaz en formato JSON⁵ o XML (eXtensible Markup Language) que le permita al cliente mantener el estado de la aplicación por su cuenta, mejorando considerablemente el rendimiento al ahorrar en ancho de banda y minimizar el tiempo que se mantiene conectada la aplicación al servidor. Por ejemplo, en una petición de datos a múltiples páginas, el cliente debería incluir el número de página a recuperar en vez de pedir "la siguiente", tal como se muestra en la ilustración 1:

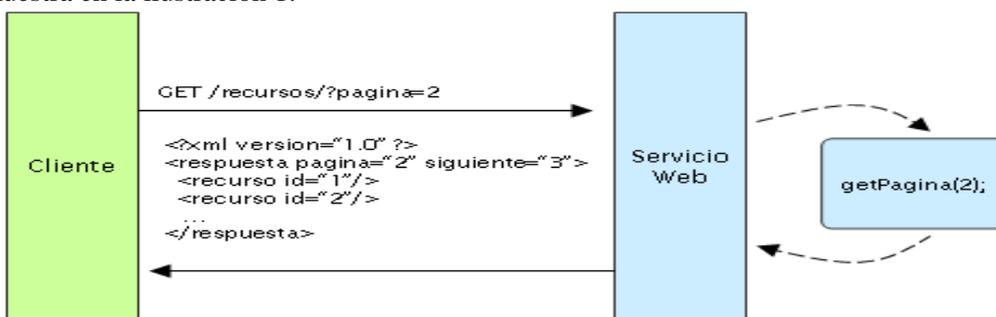


Ilustración 1. Estructura de las peticiones/respuestas al servidor utilizando REST.

³ CASE: Conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de Software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

⁴ SOAP: es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

⁵ JSON: Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Posibilita que aplicaciones con interfaces basadas en JSON o XML (AJAX) puedan conectarse, publicar y consumir recursos, lo que permite que el servicio sea utilizado por distintos clientes escritos en diferentes lenguajes, corriendo en diversas plataformas y dispositivos. Está desarrollado completamente en Java utilizando las librerías de código abierto Jasper-Reports.5.2.

El SDR es un software que permite de manera eficiente la gestión, compilación, publicación y exportación de reportes creados en herramientas de diseño de reportes como el GDR u otra aplicación que solicite el servicio, pero además permite que incluso un usuario independiente, que de forma directa interactúe con el servidor a través de algún protocolo de comunicación como el CURL puede consumir los servicios, siempre y cuando tenga los debidos permisos, recibirá exitosamente la respuesta a la petición realizada.

Para definir el mecanismo de integración se tuvo en cuenta que el motor de reportes surge para darle solución al problema específico del GDR 2.0, que es un sistema implementado en el lenguaje de programación PHP, por lo que se decidió desarrollar un API⁶ (Application Programming Interface), que constituye el intermediario entre el SDR y cualquier aplicación implementada en PHP, a través del cual resulta muy flexible la comunicación entre las aplicaciones consumidoras y el servidor, pues de modo totalmente transparente para el consumidor se realizan las peticiones y se reciben las respuestas, sin necesidad de utilizar protocolos de comunicación ni de entender la forma en que el servidor funciona, cumpliendo satisfactoriamente con los requerimientos de integración, que son críticos para construir sistemas donde los datos tienen que combinarse fácilmente y extenderse.

Como se explicó anteriormente el funcionamiento del SDR se basa en servicios web que son consumidos por las aplicaciones sin necesidad de una interfaz de usuario que medie entre los dos; no obstante, e independientemente de su funcionamiento anterior, permite funciones administrativas para la gestión y mantenimiento visual del servidor, como la gestión de usuarios y permisos, la visualización de la gestión y exportación de los reportes, la programación de tareas automáticas y la configuración visual de las políticas de seguridad de la red, por lo que también provee una aplicación intuitiva para la visualización de las funcionalidades administrativas de los recursos del servidor.

Garantiza la seguridad en el acceso a los servicios y recursos a través de un sistema de generación de token de seguridad, que consiste en una cadena de caracteres de 32 bits que se le envía al usuario luego de autenticarse con la contraseña correcta, y debe ser incluido en la cabecera de cada petición del usuario mientras el tiempo de vida del token sea válido. Además se verifica el permiso de acceso de la dirección IP (Internet Protocol) desde donde se realiza la petición, así como si el usuario tiene permisos para acceder a los recursos que solicita.

Características y funcionalidades principales:

- Permite la generación reportes en múltiples formatos como PDF, XML, HTML, CSV, XLS, RTF, TXT, JPEG, DOC, ODT, PPTX, DOCX, ODS y XSIX a partir de diferentes fuentes de datos.
- Las conexiones a las fuentes de datos de los reportes se limita solo por el soporte de los API Jdbc⁷ (Java Data Base Connection) de Java.
- Permite la generación de subreportes.
- Visualiza las vistas previas de los reportes.
- Posibilita la selección de estilos de estructura para un grupo de reportes.
- No limita el tamaño del diseño del reporte para la generación de reportes.
- Atiende peticiones de aplicaciones web o de escritorio a través de servicios REST.
- Es multiplataforma.
- Para persistir la información del servidor y los datos de los reportes, utiliza un gestor de base de datos, que puede ser PostgreSQL, SQLite, MySQL o SQL.
- Implementa un conjunto de políticas de seguridad para el acceso a los recursos a varios niveles.
- Provee un mecanismo de integración para cualquier sistema informático encargado del diseño de los reportes desarrollado en PHP, de forma que resulta transparente la comunicación entre ellos.
- Brinda una herramienta de administración visual del servidor que permite:

⁶API: conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracciones una interfaz de comunicación entre componentes de software.

⁷ Jdbc: permite la invocación de comandos SQL desde métodos de lenguaje de programación Java.

- Gestionar usuarios, permisos, conexiones a fuentes de datos y reportes.
- Visualizar cálculos estadísticos del uso de reportes.
- Programar tareas automáticas: envío de correos electrónicos con determinados reportes a un usuario específico.
- Visualizar las vistas previas de los reportes.
- Configurar visualmente las políticas de seguridad de la red.

3.3 PRUEBA DE SOFTWARE

La prueba de software es un proceso de ejecución de un programa con la intención de descubrir errores. Las pruebas definen el grado de aceptación del sistema que pueden tener los clientes del mismo, e incluso determinar si lo aceptan o no. Para ser más efectivas, las pruebas deben ser realizadas por extraños, que puedan probarlo de forma despiadada, esto garantiza pruebas con alta probabilidad de encontrar errores. Las pruebas no pueden asegurar la ausencia de errores; sólo puede demostrar que existen defectos en el software. (Pressman, 2007).

Con el objetivo de evaluar la calidad del sistema desarrollado se realizan pruebas de carga y rendimiento utilizando la herramienta informática Apache JMeter™ Desktop y pruebas de caja blanca.

Los tipos de pruebas realizadas al SDR utilizando Apache JMeter™ Desktop fueron:

- *De carga*: Prueba que se realiza para determinar y validar la respuesta de la aplicación cuando es sometida a una carga de reportes que se esperan en el ambiente de producción. Ejemplo: Comprobar la correcta respuesta del servidor ante la petición de 99 reportes en forma simultánea.

Ilustración 2. Grupos de Hilos.

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
Petición HTTP Report	99	2437	2609	3659	432	3890	0,00%	23,8/sec	3159,6
Total	99	2437	2609	3659	432	3890	0,00%	23,8/sec	3159,6

Ilustración 3. Informe Agregado: Generar 99 reportes de forma simultánea.



Ilustración 4. Gráfico de Resultados: Generar 99 reportes de forma simultánea.

Después de realizadas esta prueba, se puede concluir que el SDR está preparado para soportar la carga de múltiples peticiones simultáneas con bajo costo para el sistema y bajos tiempos de respuesta del servidor.

- *De rendimiento:* Estas pruebas se realizan para medir la respuesta del sistema informático ante distintos volúmenes de carga esperados (cantidad reportes generados). Ejemplo: Velocidad de respuesta al procesar la generación de 15, 50 y 99 reportes en forma simultánea.

Tabla 2. Valores arrojados por la prueba de rendimiento.

Reportes	Media	Mediana	Mínimo	Máximo	Rendimiento	Error
15	226	242	134ms	989ms	7,4/sec	0.00%
50	869	879	389ms	1231ms	21,0/sec	0.00%
99	1555	1370	205ms	2645ms	26,6/sec	0.00%

A partir de los valores alcanzados en las pruebas realizadas para evaluar el rendimiento del SDR con 15, 50 y 99 muestras de reportes a generarse de forma simultánea, es evidente el buen rendimiento del sistema informático desarrollado, y la capacidad de garantizar altas prestaciones en su despliegue.

4. RESULTADOS

Se obtuvo como resultado un sistema informático para la generación de reportes y subreportes de manera dinámica, y su exportación a múltiples formatos, que cuenta además con una herramienta de administración visual para la gestión de usuarios, permisos, reportes y conexiones a fuentes de datos, la exportación de los reportes, la programación de tareas automáticas, la configuración visual de las políticas de seguridad de la red, y la realización de informes estadísticos, lo cual contribuye a mejorar los tiempos de respuestas y la comunicación con el servidor; cuenta con un esquema de seguridad para el acceso a los recursos a varios niveles; y provee un método de integración con los sistemas existentes empleando nuevos paradigmas de la programación web. Permitiéndole al GDR 2.0 y a otras aplicaciones informáticas que necesiten este servicio contar con un potente y rápido motor de reportes.

A continuación se realiza una comparación de las principales características de los servidores generadores de reportes de la actualidad con el SDR (Ver la Tabla 3).

Tabla 3. Comparación de los principales motores de reportes actuales con el SDR.

Propiedades	Crystal Reports	Jaspers Server	SDR
Sistema Operativo	Windows	Multiplataforma	Multiplataforma
Fuentes de datos	Múltiples	Múltiples	Múltiples
Lenguaje de Programación	.Net	JAVA	JAVA
Servicios web	Si	Si	Si
Programación de tareas	No	No	Si
Cálculo estadístico	No	No	Si
Formatos de salida	PDF, XML, HTML, CSV, XLS, RTF, TXT, JPEG, DOC, ODT, PPTX, DOCX, ODS, XSIX	PDF, XML, HTML, CSV, XLS, RTF, TXT, JPEG, DOC, ODT, PPTX, DOCX, ODS, XSIX	PDF, XML, HTML, CSV, XLS, RTF, TXT, JPEG, DOC, ODT, PPTX, DOCX, ODS, XSIX

La tabla anterior evidencia el potencial del servidor desarrollado, que por sus características puede compararse satisfactoriamente con los servidores utilizados mundialmente y brinda un número de utilidades que incrementan su valor.

5. CONCLUSIONES

Al concluir el desarrollo de la presente investigación se puede plantear que:

- La construcción del marco teórico sobre los motores de reportes más utilizados y con mejores prestaciones permitió obtener un resultado de alto nivel que puede ser comparado satisfactoriamente con los sistemas informáticos estudiados; así como la selección correcta de la metodología de desarrollo del software, las herramientas, y tecnologías utilizadas durante el desarrollo del trabajo.
- Se implementó el sistema informático nombrado Servidor Dinámico de Reporte que posibilita la administración y exportación dinámica de reportes en múltiples formatos, además de brindar con un conjunto de servicios que facilitan la gestión de los reportes y la creación de tareas programadas como por ejemplo el envío de correos a un usuario a una determinada hora o la compilación de un reporte que se ejecute a una determinada hora del día.
- Se solucionó el problema planteado, entregando el motor de reportes imprescindible para el despliegue de la versión 2.0 del GDR, que cumple con todos los requisitos de funcionamiento definidos.
- Los resultados de las pruebas de calidad realizadas al Servidor Dinámico de Reportes demuestran la capacidad del sistema informático para garantizar altas prestaciones en su despliegue, al nivel de los motores de reportes más utilizados mundialmente.

6. REFERENCIAS

- Calzada, Leticia y José Luis Abreu. 2009. El impacto de las herramientas de inteligencia de negocios en la toma de decisiones de los ejecutivos . s.l. : International Journal of Good C, 2009. ISSN 1870-557X.
- Chen, Y. L. E. H. X. 2008. Architecture of Information System Combining SOA and BPM. s.l. : International Conference on Information Management, Innovation Management and Industri Engineering, 2008.
- Cornella, A. 1994. 'Los Recursos de Información, ventaja competitiva de las empresas', Serie McGraw Hill de Managemen. 1994.
- Cuervo, A.d. S. E. 2006. 'Nuevas Tendencias en Sistemas de Información: Procesos y Servicios', Pecvnia. 2006.
- NAN -TIC. 2013. NAN -TIC Creamos y adaptamos software libre. [En línea] 2013. [Citado el: 10 de 10 de 2013.] <http://www.nan-tic.com/es/rd/jasper-reports/>.

Pressman, Roger S. 2007. Ingeniería de Software. Un enfoque práctico. 6ta Edición. s.l. : McGraw-Hill Companies, 2007. ISBN: 8448132149.

Rodríguez, Julio César Brito. 2003. Módulo diseñador de modelos para el generador dinámico de reportes. La Habana : s.n., 2003. ISSN: 2306-2495.

SAP. 2013. The best-Run businesses Run SAP. [En línea] 2013. [Citado el: 04 de 10 de 2013.] <http://www.sap.com/solution/sme/software/analytics/crystal-bi/index.html>.