

Dynamic Report Server: Description of its Main Features

Keimer Montes Oliver, Ing.¹, Beatriz Hernández Cervantes, Ing.¹ Alberto Mendoza Garnache, Ing.¹

¹ Universidad de las Ciencias Informáticas, Cuba, kmontes@uci.cu, bcervante@uci.cu, agarnache@uci.cu

Abstract— Nowadays, the generation of reports from information stored in data sources plays an important role in decision-making processes in many entities. So, many institutions of the software industry have dedicated time and effort on implementing applications that help to improve the processing of information. Dynamic Reports Server (SDR v1.0) is a computerized system developed entirely using free software tools for management and exporting reports in various formats, extracting information from multiple data sources. Its architecture is developed in 2 layers completely oriented to REST services, making possible, thus, that services can be consumed by any software application regardless of the language in which it is developed. In the article, the theoretical and empirical methods used for the development of the tool and its main features and functionalities are listed. Besides, the different performed tests to check the required software quality for marketing it are exemplified.

Keywords— Information processing, reporting, Jasper Report., decision making.

Digital Object Identifier (DOI): <http://dx.doi.org/10.18687/LACCEI2015.1.1.034>

ISBN: 13 978-0-9822896-8-6

ISSN: 2414-6668

13th LACCEI Annual International Conference: “Engineering Education Facing the Grand Challenges, What Are We Doing?” July 29-31, 2015, Santo Domingo, Dominican Republic

Servidor Dinámico de Reportes. Descripción de sus principales funcionalidades

Ing. Keimer Montes Oliver¹, Ing. Beatriz Hernández Cervantes², Ing. Alberto Mendoza Garnache³

¹ Universidad de las Ciencias Informáticas, La Habana, Cuba, kmontes@uci.cu

² Universidad de las Ciencias Informáticas, La Habana, Cuba, bcervante@uci.cu

³ Universidad de las Ciencias Informáticas, La Habana, Cuba, agarnache@uci.cu

Abstract— Nowadays, the generation of reports from information stored in data sources plays an important role in decision-making processes in many entities. So, many institutions of the software industry have dedicated time and effort on implementing applications that help to improve the processing of information. Dynamic Reports Server (SDR v1.0) is a computerized system developed entirely using free software tools for management and exporting reports in various formats, extracting information from multiple data sources. Its architecture is developed in 2 layers completely oriented to REST services, making possible, thus, that services can be consumed by any software application regardless of the language in which it is developed. In the article, the theoretical and empirical methods used for the development of the tool and its main features and functionalities are listed. Besides, the different performed tests to check the required software quality for marketing it are exemplified.

Keywords— Information processing, reporting, Jasper Report., decision making.

Resumen:- En la actualidad, la generación de reportes a partir de la información almacenada en una fuente de datos juega un papel importante en la toma de decisiones de las entidades. Por lo que muchas instituciones de la industria del software han dedicado tiempo y esfuerzo en la implementación de aplicaciones informáticas que contribuyen a mejorar el procesamiento de la información. El Servidor Dinámico de Reportes (SDRv1.0) es un sistema informático desarrollado completamente con herramientas de software libre, para la gestión de reportes y su exportación en diversos formatos, extrayendo la información de múltiples orígenes de datos. Su arquitectura está desarrollada en 2 capas orientada completamente a servicios REST por lo que posibilita que sus servicios sean consumidos por cualquier otra aplicación informática independientemente del lenguaje en el que esté desarrollada.

En el artículo se enumeran los métodos teóricos y empíricos utilizados para el desarrollo de la herramienta, así como sus principales características y funcionalidades. Además se ejemplifican las diferentes pruebas realizadas para lograr que el software contara con la calidad requerida para su comercialización.

Palabras claves-- Procesamiento de información, reporte, Jasper Report., toma de decisiones.

I. INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TICs) juegan un papel fundamental en las empresas y ha experimentado un cambio considerable en estos últimos años, pasando de ser simples herramientas de tratamiento de datos, para convertirse en la columna

vertebral de cualquier organización tanto a nivel interno, como en lo referente a la relación con los agentes del entorno: clientes, proveedores, Administración Pública o la sociedad en general. [1]

Richard Heeks¹ señala la importancia de la adopción de las TICs para el mejoramiento de la productividad de las organizaciones, al resaltar el papel que la información juega en los procesos y en los resultados de gestión empresarial. Este autor señala la importancia de las TICs en el procesamiento de datos (i.e. convertir los datos en información valiosa para el usuario). [2]

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 en las TICs. En su estructura productiva cuenta con varios centros de desarrollo de software; uno de ellos es 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, El centro cuenta con prestigio nacional e internacional, reconocido por la integración de soluciones, productos y servicios para la captura, análisis y visualización de la información almacenada en diversos orígenes de datos.

El Servidor Dinámico de Reportes (SDRv1.0) es un sistema informático desarrollado en el centro DATEC que permite de manera eficiente la gestión, compilación, publicación y exportación de reportes. El principio del SDRv1.0 es brindar una capa de servicios que facilita su integración con cualquier otro sistema, web o de escritorio, con independencia de sus plataformas de desarrollo. A través de este los clientes pueden generar reportes en una gran cantidad de formatos sobre múltiples fuentes de datos. Estos reportes se mantendrán almacenados en el servidor, logrando con ello mejores tiempos de respuesta y acceso al historial de los reportes. Además brinda la posibilidad al usuario la creación de tareas programadas para el envío de correos electrónicos o subida a un FTP de un determinado reporte.

II. MATERIALES Y MÉTODOS

¹ Richard Heeks: Profesor de Informática en el Instituto de Desarrollo de Política de Desarrollo y Gestión de la Universidad de Manchester, Reino Unido.

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 servidores de reportes y como mejorar prestaciones del SDRv1.0. Además ayudó en el establecimiento de los métodos, herramientas y procedimientos más factibles para la descripción de los diferentes requisitos funcionales y no funcionales. 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 factible. 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.

Los métodos empíricos, como modelo de investigación científica basados en la experimentación y la lógica empírica, permitieron comprender mejor el estado del arte de las diferentes herramientas informáticas que realizan la exportación de reportes, y consecuentemente reconocer los elementos positivos que podrían tenerse en cuenta en el desarrollo de la nueva solución. Se utilizó el método experimento para realizar las pruebas y verificar el correcto funcionamiento del sistema según las normas de calidad establecidas por la universidad.

III. RESULTADO Y DISCUSIÓN

El principio del SDRv1.0 es brindar una capa de servicios que facilita su integración con cualquier otro sistema, web o de escritorio, con independencia de sus plataformas de desarrollo. A través de este los clientes pueden generar reportes en una gran cantidad de formatos sobre múltiples fuentes de datos. Estos reportes se mantendrán almacenados en el servidor, logrando con ello mejores tiempos de respuesta y acceso al historial de los reportes.

De forma general se pueden presentar las principales características del sistema de la siguiente manera:

- ✓ Permite la generación de reportes en múltiples formatos: PDF, XML, HTML, CSV, XLS, XLSX, RTF, JPG, ODT, PPTX, DOCX, ODS y XML4SWF.
- ✓ 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 REST².
- ✓ Constituye un sistema multiplataforma.
- ✓ Utiliza un gestor de base de datos, que puede ser PostgreSQL, SQLite, MySQL o SQL, para persistir la información del servidor y los datos de los reportes.

² REST: Es una técnica de arquitectura software para sistemas hipertexto distribuidos como la World Wide Web.

- ✓ Implementa un conjunto de políticas de seguridad para el acceso a los recursos y servicios a varios niveles.

De forma general se pueden presentar las principales funcionalidades del sistema de la siguiente manera:

- ✓ Permite la programación de tareas para la exportación automática de reportes en un momento específico, y su envío por correo electrónico o por FTP³.
- ✓ Brinda un conjunto de API⁴ que posibilita que la integración de las aplicaciones que utilicen el servidor sea sencilla y rápida.
- ✓ Las conexiones a las fuentes de datos de los reportes se limita solo por el soporte de los API JDBC⁵ (Java Data Base Connection) de Java.
- ✓ Permite la generación de reportes y subreportes.
- ✓ Visualiza las vistas previas de los reportes.
- ✓ Posibilita la selección de estilos de estructura para un grupo de reportes.

El SDR v1.0 tiene definida la arquitectura, sobre la base del entorno de trabajo en el que fue desarrollado como se evidencia en la Fig.1.



Fig. 1Entorno de trabajo utilizado en el desarrollo del proyecto SDR v1.0.

El SDR v1.0 brinda una capa de servicios que posibilita la integración con cualquier otro sistema, web o de escritorio, con independencia de sus plataformas de desarrollo. A través de este los clientes podrán generar reportes en una gran gama de formatos sobre múltiples fuentes de datos tales como: PostgreSQL, MySQL, Oracle y todas las que contengan un conector JDBC y sean accesibles desde el servidor. Además los clientes podrán mantener los reportes almacenados en el servidor, logrando con ello mejores tiempos de respuesta y acceso al historial de los reportes.

El funcionamiento básico del SDR v1.0 consiste en el procesamiento de los ficheros con extensión JRXML, que contienen el diseño de los reportes y pueden obtenerse por medio de las herramientas informáticas IReport o el

³ FTP: Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol).

⁴ API: Es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

⁵ JDBC: Permite la invocación de comandos SQL desde métodos de lenguaje de programación Java.

Generador Dinámico de Reportes (GDR)⁶, otro de los productos del centro DATEC especializado en el diseño de reportes; a partir de este diseño el servidor compila y genera el reporte, extrae la información de la fuente de datos específica, la incorpora al reporte y finalmente lo exporta a un formato determinado dentro de la gama de formatos soportados por el servidor. Como se muestra gráficamente en la siguiente figura, este funcionamiento se comporta de manera similar al principio mediante el cual la luz blanca se descompone en los diferentes colores del arcoíris, haciendo una analogía irreal entre la luz blanca y un fichero .JRXML, así como entre los diferentes colores y la variedad de formatos de exportación disponibles en el SDR, siendo este el prisma que media entre los anteriores elementos y produce la transformación. En la imagen también puede apreciarse una muestra de las múltiples fuentes de datos con las cuales el servidor puede establecer conexión a través de los JDBC correspondientes como se muestra en la Fig. 2.

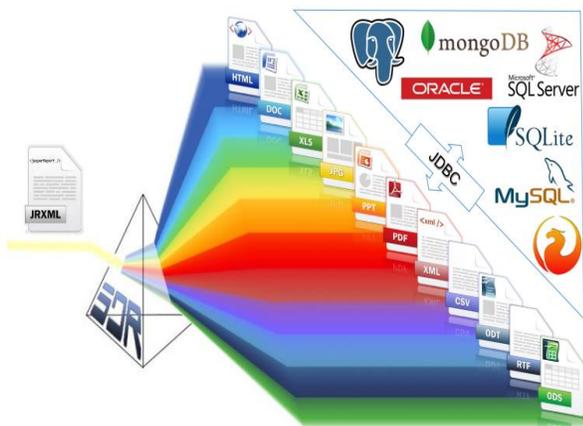


Fig. 2 Representación del Funcionamiento Lógico del Servidor Dinámico de Reportes SDR v1.0.

La arquitectura del SDRv1.0 se basa en el estilo arquitectónico en Capas, específicamente en 2 capas aplicando el estilo de Llamada y Retorno, y se utiliza la estrategia de diseño arquitectónico basada en patrones. Entre los patrones de diseño seleccionados para garantizar el éxito de la propuesta se encuentran los patrones generales de software para asignación de responsabilidades (GRASP). Se utilizaron los estándares de codificación para garantizar con su aplicación el uso mínimo de recursos en la implementación del código fuente, la rapidez en la lectura y comprensión del mismo, así como la facilidad en su depuración, mantenimiento y escalabilidad.

Estructura del SDR

El SDR v1.0 contempla el desarrollo de cinco paquetes funcionales asociados a los servicios que proveerá el mismo:

- ✓ Gestión de Usuarios y Autenticación: Permite listar, crear, actualizar, eliminar, buscar y autenticar los usuarios en el servidor.

- ✓ Gestión de Diseños de Reportes: Permite listar, buscar, publicar, actualizar, validar, compilar y eliminar diseños de reportes en el servidor.
- ✓ Gestión de Conexiones a Fuentes de Datos: Permite listar, crear, actualizar, buscar y eliminar las variables de conexión a las fuentes de datos.
- ✓ Gestión de Reportes Exportados: Permite ejecutar sobre una conexión a una fuente de datos los diseños de reportes publicados, exportándolos a los siguientes formatos (PDF, XML, HTML, CSV, XLS, XLSX, RTF, JPG, ODT, PPTX, DOCX, ODS y XML4SWF y otros soportados por la biblioteca JasperReport). Los reportes exportados se almacenan en el servidor y podrán ser consultados o eliminados con posterioridad a criterio del cliente.
- ✓ Gestión de Tareas Programadas: Permite la creación de tareas automáticas en el SDRv1.0 para el envío de las exportaciones de los reportes de forma automática a un determinado correo electrónico o su almacenamiento en un servidor FTP según la frecuencia de tiempo especificada por el cliente.

Seguridad Informática en el SDR

El proceso de gestión de la seguridad informática requiere el establecimiento de gran cantidad de controles, la implementación de variados sistemas de seguridad con mecanismos de gestión independientes, y una elevada capacidad de respuesta ante los diversos ataques y vulnerabilidades existentes; lo que hace que el mismo sea complejo y en muchas ocasiones inefectivo.

La seguridad del SDRv1.0 está basada en la Resolución 127/2007 del MIC. Esta resolución posee 100 artículos de los cuales 89 constituyen controles de seguridad informática que deben ser implementados en las instituciones. [10] Para lograr dicho objetivo el SDRv1.0 almacena la información de varios recursos con el objetivo de mantener el control de los datos imprescindibles para el procesamiento de los servicios y para la exportación de los reportes almacenados.

Estos recursos son los que a continuación se listan:

- ✓ Usuarios (SysUser): Constituyen los clientes del servidor, son los encargados de realizar las peticiones y generalmente serán otras aplicaciones informáticas. Nombre del recurso (resource) para las peticiones: user.
- ✓ Conexiones a Bases de Datos (DbConnection): Son las conexiones a las fuentes de datos de los reportes, donde se encuentra almacenada la información que será cargada en los reportes y mostrada en las exportaciones. Nombre del recurso (resource) para las peticiones: connection.
- ✓ Reportes (Report): Son los diseños de los reportes enviados por los usuarios al servidor. Nombre del recurso (resource) para las peticiones: report.

⁶ GDR: Es una aplicación desarrollada en la UCI para el diseño de reportes en la web.

- ✓ Exportaciones (Export): Son las exportaciones realizadas a los reportes a un formato específico. Nombre del recurso (resource) para las peticiones: export.
- ✓ Tareas Programadas (Scheduled): Constituyen tareas que se programan para la exportación automática de reportes en un determinado momento y su envío por correo electrónico o por FTP. Nombre del recurso (resource) para las peticiones: task.

Para garantizar la seguridad en el acceso a los recursos y a los servicios que brinda el SDRv1.0 cuenta con un sistema de generación de token, que consiste en una cadena alfanumérica de 32 caracteres 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. Cuando un token de autenticación expira es eliminado de la base de datos y no podrá ser utilizado en ninguna petición.

Antes de proceder con la asignación de un token al usuario además se verifica el permiso de acceso de la dirección IP (Internet Protocol) desde donde se realiza la petición en correspondencia con la máscara de red especificada para el usuario en el servidor; además se verifica si el rol que tiene asignado el usuario tiene acceso al recurso solicitado cada vez que se consume un servicio.

Cada petición que realice un usuario debe incluir token en la cabecera de la URL⁷ para poder acceder a los servicios porque siempre se verifican las restricciones de seguridad explicadas anteriormente, en caso que el si el resultado de la verificación es insatisfactoria el token de autenticación enviado será eliminado y no podrá volver a ser utilizado en otra petición.

Formas de consumir los servicios

Al basarse el SDRv1.0 en servicios web estos pueden ser consumidos por las aplicaciones clientes sin necesidad de una interfaz de usuario que medie entre los sistemas implicados. No obstante, e independientemente de su funcionamiento anterior, es posible que un usuario interactúe de forma directa con el servidor a través de un protocolo de comunicación, esto puede lograrse, por ejemplo, utilizando las facilidades brindadas por la librería CURL⁸, la cual fue utilizada para ejemplificar el funcionamiento de los servicios brindados por el SDRv1.0.

Para la realización de las peticiones utilizando CURL se siguen los siguientes pasos:

1. Se crea un fichero de texto con la extensión .sh si se encuentra en el sistema operativo Linux o .bash en el sistema operativo Windows.

2. Dele permisos de ejecución en las propiedades del fichero.

3. Copie para el fichero, del contenido del servicio deseado, la sección titulada Ejemplo de petición. (Para poder ejecutar exitosamente una petición, el primer servicio que debe solicitar es el de autenticación para que le sea asignado un token con el cual podrá solicitar otros servicios).

4. Debe modificar la petición, cambiando el token de autenticación que se encuentra en el ejemplo por el que le fue asignado en la reciente autenticación, este elemento se determina en la línea -H 'X-SDR-API-Key: TOKEN DE AUTENTICACIÓN'.

5. Si la petición requiere de parámetros específicos debe proporcionarlos correctamente para recibir una respuesta satisfactoria.

6. Guarde los cambios efectuados en el fichero .sh o .bash.

7. Abra la terminal de comandos del sistema operativo y acceda al directorio donde creó el fichero .sh o .bash. Ejecute el fichero y espere la respuesta del servidor. En la Figura 5 se muestra visualmente el procedimiento realizado en el sistema operativo Linux, y se resalta la sección de la respuesta del servicio de autenticación donde aparece el token asignado al usuario, esta cadena es la que debe ser enviada en la cabecera de cada petición. El servidor cuenta con su propia estructura interna para retornar la respuesta al cliente de un modo especificado por él (JSON⁹ o XML), esta se compone de un conjunto de valores que muestran el resultado de la petición, donde los elementos tienen el siguiente significado:

- ✓ **successful:** es un valor booleano (true/false) que indica si la petición se realizó satisfactoriamente o no.
- ✓ **errorMsg:** en caso de que la petición no se realice satisfactoriamente muestra el mensaje del error acontecido.
- ✓ **exception:** en caso de que la petición no se realice satisfactoriamente muestra el tipo de error acontecido.
- ✓ **totalItems:** en caso de que la petición se realice satisfactoriamente muestra la cantidad de elementos contenidos en la respuesta.
- ✓ **items:** en caso de que la petición se realice satisfactoriamente contiene una lista con los elementos enviados en la respuesta.
- ✓ **url:** este elemento solo se utiliza en la exportación de reportes, que en caso de que la petición se realice satisfactoriamente contiene la dirección URL con la cual el usuario puede acceder directamente al reporte exportado.

⁷ URL: Cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en la Internet.

⁸ CURL: Es una herramienta para usar en un intérprete de comandos para transferir archivos con sintaxis URL, soporta FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, FILE y LDAP.

⁹ 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.

Se obtiene una respuesta insatisfactoria cuando ha ocurrido algún error en el procesamiento de la petición, debido tanto a factores internos del servidor como externos al mismo, como puede ser el fallo en la conexión a las fuentes de datos de los reportes o a los servidores FTP.

Ejemplo de respuesta satisfactoria:

```
{ "successful":true,"errorMsg":"","exception":"","totalItems":1,"items":[{"id":"6918E5127859474EB7348CF6DFCC05110","format":"pdf","runtime":3481,"remoteIp":"127.0.0.1","createdAt":1409856668132,"status":"OK","params":{"name":"IDS","value":"2013","type":"String"},"reportId":19851,"dbConnectionId":6502,"dbConnection":{"id":6502,"driver":"org.postgresql.Driver","name":"Connection","password":"****","url":"jdbc:postgresql://127.0.0.1:5432/sdr","username":"postgres"},"pathExport":"19851"}]}
```

Ejemplo de respuesta insatisfactoria:

```
{ "successful": false,"errorMsg":"ERROR (107): The resource was not found.","exception":"sdr.exception.ValidationException","totalItems":0,"items":[] }
```

Tipos de Pruebas aplicadas al SDR

A continuación se describen los tipos de pruebas aplicadas durante las Pruebas de Liberación, los cuales fueron definidos en función de las características de calidad identificadas en la norma ISO/IEC 9126-1 Parte 1: Modelo de Calidad, homologada en Cuba, y teniendo en cuenta las sub-características, las cuales han sido mejoradas en la norma ISO/IEC 25010, Software Product Quality Requirements and Evaluation (SquaRE).

Funcionalidad: Consisten en la revisión de los requisitos aceptados por el cliente contra las funcionalidades presentes en la aplicación.

Seguridad: Asegurar que los datos o el sistema solamente es accedido por los actores definidos según niveles de acceso. En 3 niveles.

Pruebas de Rendimiento: Enfocadas a monitorear el tiempo en flujo de ejecución, acceso a datos, en llamada a funciones y sistema para identificar y direccionar los cuellos de botellas y los procesos ineficientes.

Pruebas de Carga: Usada para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.

Pruebas de Estrés: Enfocada a evaluar cómo el sistema responde bajo condiciones anormales. (Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible).

Una vez comprobada la funcionalidad del SDRv1.0 y su capacidad para asumir la carga de múltiples peticiones

concurrentes manteniendo un buen rendimiento, es preciso compararlo con los servidores de reportes más utilizados mundialmente, en la Tabla 1 muestra brevemente el resultado de la comparación a través de los criterios comparativos trascendentales para la investigación, donde se evidencia que el servidor desarrollado con las herramientas libres puede compararse satisfactoriamente con el Crystal Reports y el Jaspers Server.[3][5]

Tabla 1 Coparación entre los principales Servidores de Reportes

Propiedades	Crystal Reports	Jaspers Server	SDRv1.0
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	Sí
API para la integración de servicios	No	No	Si

IV. CONCLUSIONES

- ✓ A partir de las funcionalidades identificadas se realizó el modelo de diseño basado en la arquitectura 2 capas, proporcionando la entrada apropiada y el punto de partida para las actividades de implementación.
- ✓ Se implementó la herramienta multiplataforma que posibilita la generación de reportes extrayendo la información de diferentes fuentes de datos de forma rápida y sencilla
- ✓ Se suministra a las aplicaciones clientes un API para mejorar la integración con el SDR sin importar el lenguaje implementado.
- ✓ Las pruebas realizadas para validar las funcionalidades implementadas, demostraron que se desarrolló una herramienta con la calidad requerida.

REFERENCIAS

[1] C. S. R. Álvaro Gómez Vieites, Sistemas de Información. Herramientas prácticas para la gestión., RA-MA S.A., Oct 9, 2009.

[2] R. Monge Gonzalez y C. Alfaro Azofeita, TICs en Las PYMES de Centroamérica: Impacto de la Adopción de las tecnologías de la información y la comunicación en el desempeño de las empresas, Costa Rica: Editorial Tecnológica Costa Rica, 2005.

[3] Crystal Solutions, «SAP Crystal Solutions,» [En línea]. Available:

- <http://www.sap.com/solution/sme/software/analytics/crystal-bi/index.html>. [Último acceso: 8 1 2015].
- [4] R. S. Presman, Ingeniería de Software. Un enfoque práctico., 6 ed., McGraw-Hill Companies, 2007.
- [5] «NAN -TIC Creamos y adaptamos software libre.» [En línea]. Available: <http://www.nan-tic.com/es/rd/jasper-reports/>. [Último acceso: 15 1 2015].
- [6] J. L. A. Leticia Calzada, El impacto de las herramientas de inteligencia de negocios en la toma de decisiones de los ejecutivos, International Journal of Good, 2009.
- [7] S. O. Bonilla, «Ingeniería de Sistemas.» [En línea]. Available: <http://ingenieriasystem-sabados.blogspot.com/>. [Último acceso: 15 01 2015].
- [8] M. I. A. G. Sommerville, Ingeniería del software, Madrid: Pearson Educación. S.A, 2006.
- [9] S. Izza, Integration of industrial information systems: from syntactic to semantic integration approaches, Enterprise Information Systems, 2009.
- [10] MIC, Resolución 127/2007 MIC. Reglamento de seguridad para las tecnologías de la información, Ministerio de la informática y las comunicaciones (MIC), 2007.
- [11] S. Radack and R. Kuhn, Managing Security: The Security Content Automation Protocol, IT Professional, 2011.