Authentication Component for Dynamic Report Generator

Miguel Lezcano Ramos, Ing.[1], Orestes Febles Díaz, DrC.[1]
[1]Universidad de las Ciencias Informáticas, Cuba, mlezcano@uci.cu, ofebles@uci.cu

Abstract– ***The Dynamics Report Generator (GDR from the Spanish long) is a platform for the reports management. This tool allows to build reports based on information stored in different Database Management Systems. It also offers a flexible reports design oriented to the end users, highlighting the Drag and Drop options. Its functionalities have been combined with several management systems to facilitate the reuse of these ones and the resources rationalization. Some scenarios related to centralized authentication have appeared in some of the places where the system has been deployed, and though the system has a satisfying performance and meets the overall requirements for a reports generating system, it is not prepared to be incorporated to this type of scenarios. Analyzing this need, it was undertook an investigation related to the latest tendencies in the web applications development and the incorporation of centralized authentication mechanisms to obtain an authentication component for the GDR which includes centralized authentication. This investigation aimed to develop an authentication component in the GDR security module, including the necessary setting options to guarantee that the system adapts to the authentication scenarios identified in the places it has been deployed. The development of the component was driven by the OpenUp development methodology. It was successfully tested in a centralized authentication environment and an installation guide was produced to embed the component in every organization using the system.***

Keywords-- GDR, authentication, SSO, CAS.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2015.1.1.020
ISBN: 13 978-0-9822896-8-6
ISSN: 2414-6668

Componente de autenticación para el Generador Dinámico de Reportes

Miguel Lezcano Ramos, Ing.[1], Orestes Febles Díaz, DrC.[1]
[1]Universidad de las Ciencias Informáticas, Cuba, mlezcano@uci.cu, ofebles@uci.cu

Resumen– ***El Generador Dinámico de Reportes (GDR) es una plataforma para la gestión de reportes. Esta herramienta permite construir informes partiendo de la información almacenada en diferentes gestores de bases de datos, además de brindar un diseño flexible de los reportes y orientado al usuario mediante las opciones de Drag and Drog. Sus funcionalidades han sido combinadas con varios sistemas de gestión, facilitando la reutilización de las mismas y la racionalización de recursos. En algunas de las organizaciones donde se ha desplegado el sistema aparecen diversos escenarios relacionados con la autenticación centralizada. Aunque el sistema mantiene un rendimiento satisfactorio y cumple con los requisitos generales de un sistema para la generación de reportes, no se encuentra preparado para incorporarse a tales escenarios. A partir de esta necesidad se realizó un estudio relacionado con las últimas tendencias en el desarrollo de aplicaciones web y la incorporación de mecanismos de autenticación centralizada. La presente investigación tuvo como objetivo desarrollar un componente de autenticación en el módulo de Seguridad del GDR con las opciones de configuración necesarias para garantizar que el sistema se adapte a los escenarios de autenticación identificados en las organizaciones donde ha sido desplegado. El desarrollo del componente estuvo guiado por la metodología de desarrollo OpenUp. Fue probado en un ambiente de autenticación centralizada con resultados satisfactorios y se elaboró una guía para incorporar el componente en las demás organizaciones donde está desplegado el sistema.***

Palabras claves-- GDR, autenticación, SSO, CAS.

Abstract– ***The Dynamics Report Generator (GDR from the Spanish long) is a platform for the reports management. This tool allows to build reports based on information stored in different Database Management Systems. It also offers a flexible reports design oriented to the end users, highlighting the Drag and Drop options. Its functionalities have been combined with several management systems to facilitate the reuse of these ones and the resources rationalization. Some scenarios related to centralized authentication have appeared in some of the places where the system has been deployed, and though the system has a satisfying performance and meets the overall requirements for a reports generating system, it is not prepared to be incorporated to this type of scenarios. Analyzing this need, it was undertook an investigation related to the latest tendencies in the web applications development and the incorporation of centralized authentication mechanisms to obtain an authentication component for the GDR which includes centralized authentication. This investigation aimed to develop an authentication component in the GDR security module, including the necessary setting options to guarantee that the system adapts to the authentication scenarios identified in the places it has been deployed. The development of the component was driven by the OpenUp development methodology. It was successfully tested in a centralized authentication environment and an installation guide was produced to embed the component in every organization using the system.***

Keywords-- GDR, authentication, SSO, CAS.

INTRODUCCIÓN

Las cuestiones de seguridad relacionadas con la tecnología de la información siguen siendo una preocupación en la sociedad actual, y para quienes toman decisiones sobre la misma [1]. La creciente sofisticación de las amenazas a la seguridad de la información ha hecho de ésta una función crítica en muchos sectores de negocio [2].

En la actualidad las empresas manejan grandes volúmenes de información. Estas necesitan tener almacenados todos los datos concernientes a sus negocios en bases de datos, para gestionarlos mediante una aplicación profesional. Sin esta funcionalidad, resultaría imposible manejar en su totalidad la información que se genera en la empresa, ocasionando pérdidas de tiempo y dinero [3].

Como medio primario para sintetizar la información esencial en apoyo a la toma de decisiones, se aprovechan los reportes, formados mediante la combinación lógica de datos generalmente dispersos. Seleccionando, organizando y estableciendo relaciones entre dichos datos de forma adecuada, se obtienen resultados precisos para las personas apropiadas dentro de empresas u organizaciones [4].

Existen diferentes tecnologías que permiten generar reportes de forma dinámica y así lograr una mayor efectividad a la hora de hacer uso de ellos. Estas tecnologías son poderosos motores de reportes que pueden generar informes resumidos y detallados en tiempo real desde diferentes bases de datos [5].

Una de las herramientas que hace uso de estas tecnologías es el GDR, desarrollado por DATEC (Centro de Tecnologías de Gestión de Datos) perteneciente a la UCI (Universidad de las Ciencias Informáticas). El GDR es una herramienta web que gestiona el ciclo de vida de los reportes. Accede a la información almacenada en los gestores de base de datos PostgreSQL, MySQL, SQLite, Oracle y MS-SQL Server. Permite un diseño flexible de los reportes y orientado al usuario, haciendo uso de las opciones de *Drag and Drog*.

Los autores consultados en la referencia [6] coinciden en que el GDR es una solución abierta y extensible para los informes administrados. Su arquitectura flexible permite a los programadores de software y a las empresas integrar el conjunto de herramientas con sus sistemas heredados, portales empresariales o aplicaciones personalizadas.

El sistema cuenta con varios módulos, entre ellos el de Seguridad. Según la referencia [7] de la bibliografía consultada, entre sus funcionalidades se encuentra la de gestionar los usuarios que se encuentren local o en uno o varios LDAP previamente configurados en la vista Autenticación LDAP. Esto le permite al GDR contar con

autenticación local y por LDAP para así restringir el acceso a usuarios autorizados.

Sin embargo el GDR no se encontraba preparado para asumir todos los escenarios de autenticación identificados en las organizaciones que utilizan dicho sistema. Ente estos escenarios se observa de manera común el de autenticación centralizada. Una de las soluciones adoptada era inhabilitar la autenticación del GDR y delegar este proceso a la aplicación de gestión con la cual se integraba, requiriendo más seguridad a nivel de servidores y de redes para mitigar el acceso a funcionalidades determinadas por usuarios no autorizados.

Otra alternativa era combinar los códigos del GDR con la aplicación con la que se integraba convirtiéndose en un solo sistema y así compartir la misma autenticación, pero para esto debían compartir las mismas tecnologías, además de que el mantenimiento al GDR se hacía bastante engorroso. Teniendo en cuenta los elementos planteados se realizó una investigación para una posterior modificación del sistema la cual permitiera al GDR contar con autenticación centralizada.

Teniendo en cuenta los elementos planteados surge la necesidad de realizar una investigación para incorporar al sistema las opciones necesarias para configurar los escenarios de autenticación identificados en las organizaciones donde ha sido desplegado. La solución propuesta consiste en el desarrollo de un componente de autenticación en el módulo de Seguridad del GDR aplicando patrones de diseño y elementos de la programación orientada a componentes, para garantizar que el GDR se ajuste a los escenarios de autenticación identificados y para futuros escenarios que puedan incorporarse a posteriori.

MATERIALES Y MÉTODOS

La seguridad de la información es la protección de los sistemas de información, y de la información en sí, del acceso, uso, modificación o destrucción no autorizado, a fin de garantizar la confidencialidad, integridad y disponibilidad [8]. La seguridad de la información cuenta con aspectos funcionales como son la autenticación, la autorización, la confidencialidad de los datos, la integridad de los datos, la protección contra ataques continuos y la protección de la privacidad y aspectos no funcionales como son la interoperabilidad, manejabilidad y facilidad de desarrollo [9].

Para garantizar la seguridad de la información es necesario contar con un mecanismo de control de acceso. El autor de la bibliografía consultada en la referencia [10] afirma que el control de acceso es un aspecto de seguridad cuyos requisitos evolucionan con los avances tecnológicos y al mismo tiempo, con los contextos sociales contemporáneos. Por otra parte los autores en la referencia [11] plantean que el control de acceso es indispensable en las organizaciones cuyo funcionamiento requiere el intercambio de recursos digitales con diversos grados de sensibilidad.

En la literatura consultada la mayoría de los autores coinciden que el control de acceso debe integrar los procesos de identificación, autenticación, autorización y auditoría con el objetivo de verificar, conceder, denegar y auditar el acceso de los usuarios a los recursos gestionados con el uso de las Tecnologías de la Información y la Comunicación [12]. En la Fig. 1 se muestra el flujo básico para establecer el control de acceso sobre un recurso determinado.

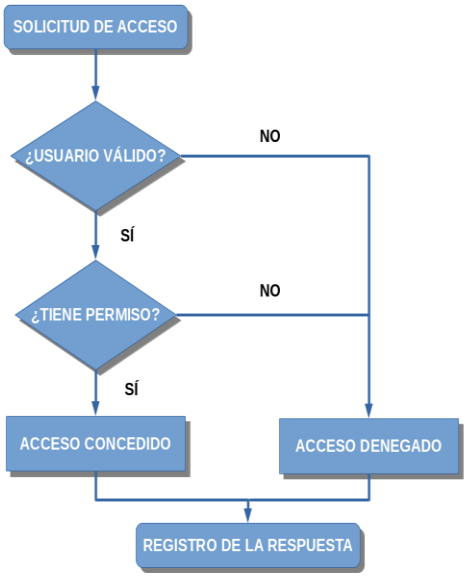

Fig. 1 Flujo de control de acceso. Fuente: Elaboración propia.

El proceso de autenticación es la verificación de que un usuario que intenta identificarse es válido, usualmente se implementa con una contraseña en el momento de iniciar una sesión [13]. En la actualidad, los sistemas de información incorporan módulos específicos de identificación y autenticación de usuarios. Esto trae consigo que la información del usuario pueda ser diferente en cada uno de los sistemas y que tenga que recordar credenciales de acceso diferentes para cada uno de ellos [14]. Para evitar esto surge la autenticación centralizada (*Single Sign-On SSO*). Ésta no es más que un mecanismo que permite a un usuario acceder a varias aplicaciones autenticándose una sola vez.

El principio de todas las soluciones SSO es eliminar la autenticación de todas las aplicaciones. Para esto se necesita un servidor de autenticación que reciba las credenciales de un usuario, redireccionamiento HTTP de las aplicaciones al servidor de autenticación para usuarios no registrados y de vuelta a dichas aplicaciones cuando se encuentre autenticado y que la información sea pasada por el servidor de autenticación hacia las aplicaciones durante el redireccionamiento, por medio de las *cookies* [15].

Cada vez son más las aplicaciones que permiten delegar la autenticación de sus usuarios. Esta opción resulta muy cómoda puesto que se reduce el número de credenciales a administrar y se evitan los procesos relacionados con la gestión de usuarios, además de propiciarle a los usuarios una experiencia sencilla para iniciar su sesión una sola vez para acceder a todos sus

recursos sin tener que compartir inadvertidamente información confidencial con los usuarios no autorizados.

Uno de los ejemplos más ilustrativos de la aplicación de SSO es la plataforma de aplicaciones de Google (*Google Apps*). Esta plataforma ofrece un servicio de inicio de sesión único a clientes que usan *Google Apps for Business*, *Google Apps for Education o Google Apps for ISPs*. Mediante este servicio, los usuarios pueden acceder a todas las aplicaciones que brinda dicha plataforma (incluida la Consola del administrador) con solo iniciar sesión una vez en una página de SSO. Google redirige a los usuarios a la página de SSO cuando intentan iniciar sesión en la Consola del administrador o en otro servicio de Google [16]. En la Fig. 2 se muestra el formulario de inicio de sesión única brindada por la plataforma de aplicaciones de Google cumpliendo con los principios del SSO para acceder a cualquiera de sus aplicaciones.

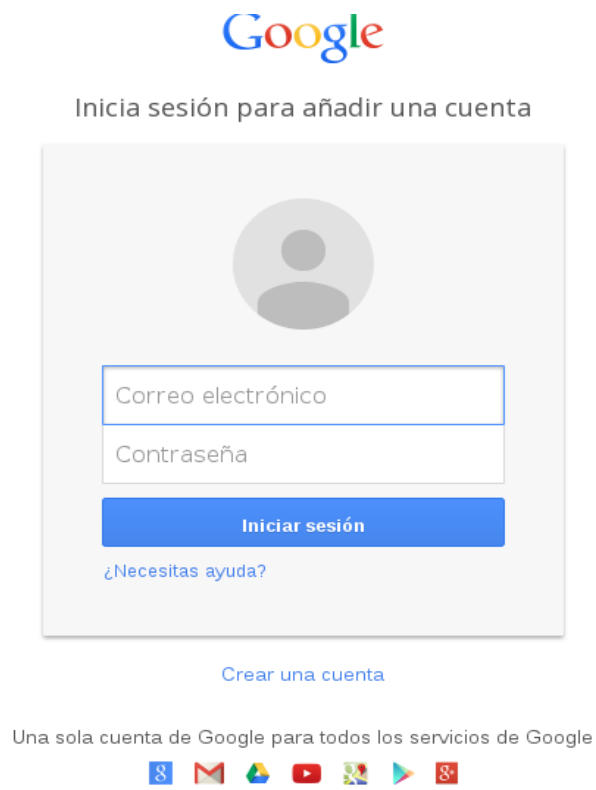

Fig. 2 Formulario de inicio de sesión único de la plataforma de aplicaciones de Google. Fuente: https://accounts.google.com/AddSession.

Este servicio está basado en SAML 2.0. Usando este modelo, dicha plataforma actúa como proveedor de servicios (Gmail, Mapa, Calendario, Traductor) y como proveedor de identidad, controlando las credenciales de usuarios que intentan acceder a cualquiera de estos servicios. Este servicio solamente puede ser consumido por aplicaciones web. En la Fig. 3 se muestra la implementación de SAML-SSO para la plataforma de aplicaciones de Google.

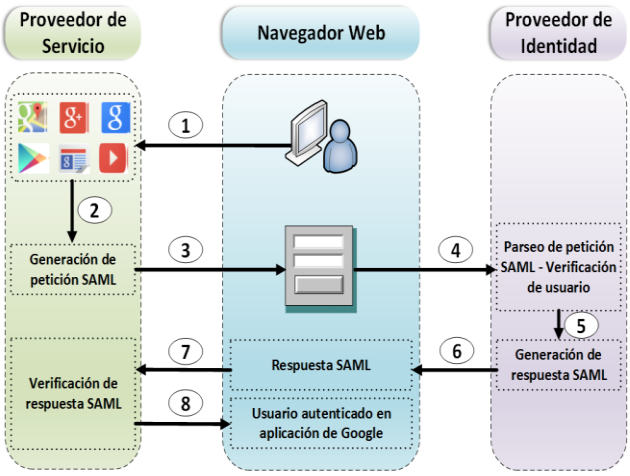

Fig. 3 SSO-SAML Google. Fuente: Elaboración propia.

Cuando un usuario intenta acceder a una aplicación de la plataforma, Google genera una solicitud de autenticación SAML y envía al usuario para el formulario de inicio de sesión. Una vez introducidas las credenciales, son enviadas al proveedor de autenticación, el cual parsea la solicitud de inicio de sesión SAML y verifica la existencia de dicho usuario. Posteriormente genera una respuesta SAML codificada que se verifica en el proveedor de servicios para permitir o denegar el acceso a la aplicación solicitada por el usuario.

De igual manera existen aplicaciones web que implementan SSO. Entre ellas se encuentra el Servicio de Autenticación Central (*Central Authentication Service CAS*). En la referencia [17] de la bibliografía consultada el autor afirma que el CAS es un producto multiprotocolo web de inicio de sesión única (SSO) integrado por un único componente de servidor lógico que realiza las solicitudes de autenticación de varios clientes que se comunican a través de uno o más protocolos soportados.

Según el colectivo de autores en la referencia [18] existen varias implementaciones del servidor CAS de código abierto. Entre ellos la propuesta desarrollada por el Grupo de Interés Especial en Arquitecturas Java (*Java Architectures Special Interest Group JASIG*) es el más utilizado por aplicaciones empresariales.

Básicamente cuando un usuario solicita una petición a una aplicación web suscrita al CAS, este verifica si está autenticado. Si la autenticación no se ha realizado automáticamente lo envía para el formulario de inicio de sesión del CAS. Una vez realizada la autenticación correctamente el CAS vuelve a redirigir a la petición que se ejecutó en un primer momento. Esto evita a los desarrolladores contar con una formulario de inicio de sesión por cada aplicación, basta con comprobar si el usuario que intenta acceder ya ha sido registrado. En la Fig. 4 aparece un esquema que refleja en detalles el funcionamiento del CAS.

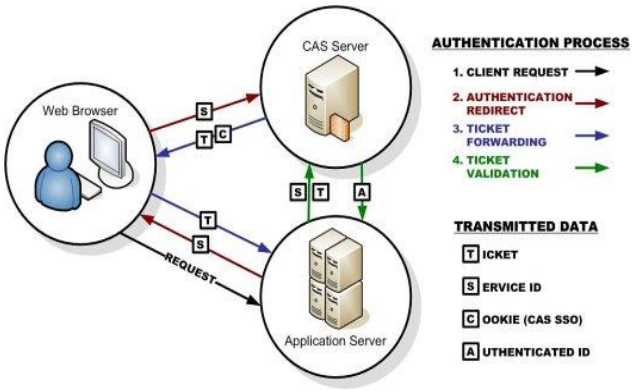

Fig. 4 Funcionamiento del CAS. Fuente: [19].

RESULTADOS Y DISCUSIÓN

Para el desarrollo del componente se utilizó el lenguaje de programación PHP para la capa del servidor debido a que tiene soporte para la mayoría de los servidores web en la actualidad como son: Apache, *Microsoft Internet Information Server*, *Personal Web Server*, *Nginx* y *Lighttpd*. Además puede valerse en los principales sistemas operativos como son: Linux, Microsoft Windows y Mac OS X.

Como marco de trabajo se escogió Symfony 1.4.20, para gestionar todas las peticiones que lleguen al servidor de la aplicación. Para el desarrollo de las interfaces de usuario se emplea ExtJS por la similitud de sus componentes a los de las aplicaciones tradicionales de escritorio. Como herramienta de modelado se escogió *Visual Paradim* por ser multiplataforma y su facilidad para generación de código en múltiples lenguajes. Para guiar todo el proceso de desarrollo se optó por la metodología de desarrollo *OpenUp* ya que incorpora técnicas ágiles probadas para la construcción de sistemas con alta calidad.

Debido a sus características, el GDR, ha sido integrado con varias soluciones para la generación de los reportes en las siguientes organizaciones cubanas: Oficina Nacional de Estadística e Información (ONEI), Contraloría General de la República (CGR), Fiscalía General de la República (FGR), Comisión Electoral Nacional (CEN), Tribunal Supremo de Justicia (TSP) y Dirección General de Proyectos de la UCI (DGP).

También ha sido desplegado en instituciones extranjeras como son: Instituto Nacional de Gestión de Bolsas de Estudio de Angola (INAGBE), Sistema Autónomo de Registros y Notarías de Venezuela (SAREN), Misión Vivienda de Venezuela, Ministerio del Poder Popular para la Energía y el Petróleo de Venezuela (MENPET) y Ministerio del Poder Popular para Relaciones Interiores, Justicia y Paz de Venezuela (MPPRIJ).

En el caso de este último, por su importancia y requerimientos de seguridad fue donde primero se aplicó el componente. Se contaba con balanceadores de carga para el portal web (SINASEC), el GDR y el CAS, además de un *cluster* de bases de datos donde se encontraba la información consultada para la creación de los reportes. SINASEC fue desarrollado utilizando Liferay 6.0.5 basado en JAVA, el cual estaba encomendado a gestionar todos los procesos requeridos por la institución. Por otra parte el GDR estaba encargado de mostrar todos los reportes que eran solicitados a través del portal. En la Fig. 5 se muestra la vista que integra ambos sistemas haciendo uso de la autenticación centralizada por medio del CAS.

Muchas de las opciones brindadas por SINASEC estaban relacionadas a la visualización de información en forma de reporte, lo cual requería un uso sostenido del GDR. Mediante la integración con el CAS se garantizaba que tanto en SINASEC como en GDR el usuario autenticado fuese el mismo.

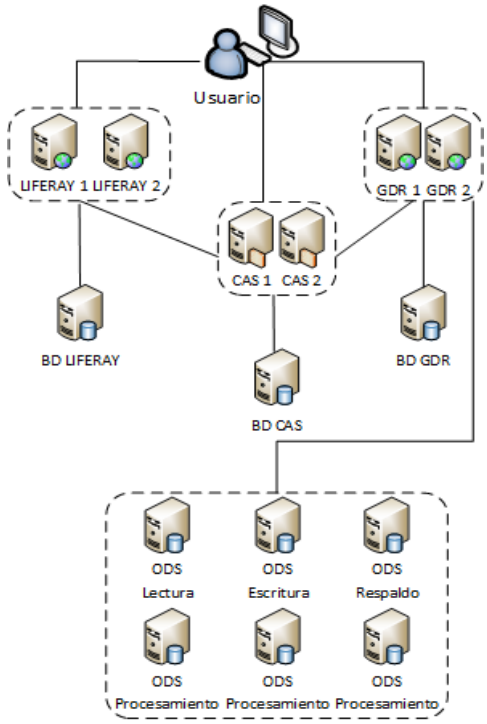

Fig. 5 Vista integrada. Fuente: Elaboración propia.

Con vista a soportar SSO implementado por el CAS desde el GDR, se adoptó un nuevo diseño de clases para el componente de autenticación como se muestra en la Fig. 6.

En este diseño se tuvieron en cuenta la autenticación local y por LDAP los cuales ya estaban soportados con anterioridad por el GDR y se incorporó la autenticación centralizada a través del CAS. Gracias a que el CAS cuenta con librerías implementadas en varios lenguajes de programación, entre ellas PHP, facilitó la incorporación del GDR a un escenario de SSO. La librería utilizada para comunicar al GDR con el CAS fue phpCAS. Además con este diseño de clases se garantiza que arquitectónicamente sea extensible para otras formas de autenticación que puedan incluirse a posteriori.

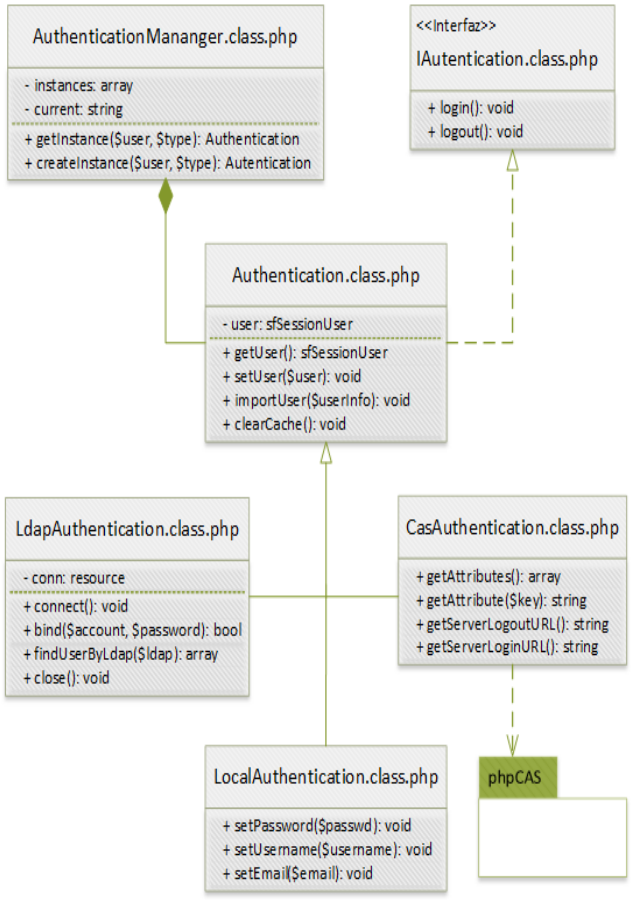

Fig. 6 Diagrama de clases para los tipos de autenticación. Fuente: Elaboración propia.

La interfaz IAutentication.class.php es la que define los servicios para la realización del inicio (login) y cierre (logout) de sesión según las características de cada modo de autenticación incorporado. La clase Authentication.class.php contiene las funcionalidades comunes que requiere cada tipo de autenticación. La clase LocalAuthentication.class.php permite realizar la autenticación de forma local validando las credenciales introducidas en el formulario de inicio de sesión del GDR por medio de la tabla tbuser de su base de datos. Esta clase incorpora funcionalidades particulares para el cambio de la contraseña, nombre de usuario y dirección de correo electrónico de un usuario previamente seleccionado. La clase LdapAuthentication.class.php al igual que la autenticación local muestra el formulario de inicio de sesión del GDR, pero delega el proceso de validación de las credenciales introducidas contra uno de los LDAP que se encuentran definidos en el sistema (ver Fig. 7) en caso de que el usuario ya se encuentre registrado en la tabla tbuser del GDR. En caso contrario el sistema buscará dicho usuario por cada uno de los LDAP que se hayan configurado. Si el mismo es encontrado y sus credenciales son correctas se registra en la tabla tbuser y se referencia a que LDAP pertenece para una futura autenticación. La clase CasAuthentication.class.php redirecciona al formulario de inicio de sesión del CAS propiciando la autenticación centralizada. La clase AuthenticationMananger.class.php gestiona las instancias de los tipos de autenticación para acceder a sus funcionalidades.

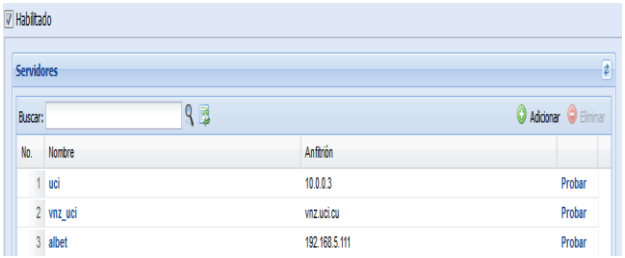

Fig. 7 Listado de LDAP disponibles.

La información requerida para integrarse con el CAS es recogida a través de un formulario como se muestra en la Fig. 8. El CAS establece como política para asegurar el SSO que el formulario de inicio de sesión este publicado bajo el protocolo HTTPS. Esto cerciora que las credenciales que se envían al servidor de autenticación viajen de forma segura. Por lo tanto es necesario tener en cuenta en el formulario qué certificado se usará. La casilla de selección *"Habilitado"* le dirá al GDR si debe o no delegar la autenticación a un servidor CAS. Una vez cambiado este estado, el sistema notifica que los cambios tendrán efecto cuando se cierre la sesión actual. Con esto se logró poder configurar visualmente y orientado al usuario el modo de autenticación mediante el CAS desde el GDR.

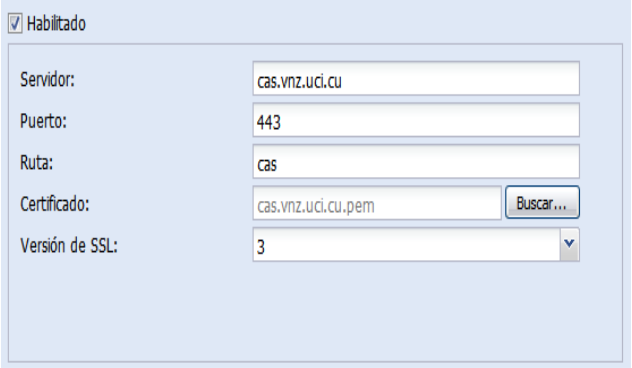

Fig. 8 Formulario para la integración con el CAS.

El componente de autenticación en el escenario MPPRIJ, fue probado siguiendo la metodología de software adoptada mediante la ejecución de pruebas funcionales e integración al sistema para garantizar el correcto funcionamiento entre el GDR, el CAS y SINASEC. Además se realizó una prueba de aceptación en la cual los clientes verificaron que dicho componente cumple con los requisitos de seguridad requeridos por la institución. Después de obtener los resultados esperados como se muestra en la Fig. 9, se procedió a la elaboración de una guía para la actualización del GDR en las restantes organizaciones donde se encuentra desplegado generalizando el aporte obtenido con este componente de autenticación.

Fig. 9 Visor de reportes del GDR integrado con SINASEC.

CONCLUSIONES

La autenticación centralizada permite a los usuarios acceder a varias aplicaciones introduciendo sus credenciales una sola vez por medio de un formulario de autenticación único. Esto evita las múltiples administraciones de cuentas de usuario y el acceso no autorizados si la seguridad de alguna de las soluciones es quebrantada.

El CAS es una solución libre, robusta y probada para escenarios de autenticación centralizada, que suscribe varias aplicaciones compartiendo un formulario de autenticación común. El estudio de esta alternativa fue tomada en cuenta para dar respuesta a las carencias del GDR frente a escenarios con autenticación compartida.

La implementación del componente fue probado en el escenario MPPRIJ con resultados satisfactorios cumpliendo con los requisitos pactados con la institución. Aportó al GDR una vía de autenticación centralizada mediante el CAS y la posibilidad de incorporar otros escenarios a posteriori. Para generalizar el aporte obtenido se elaboró una guía de actualización con el objetivo de ser aplicada en las restantes entidades donde se está utilizando el GDR.

REFERENCIAS

[1] T. Sommestad, M. Ekstedt, and P. Johnson. (2010, A probabilistic relational model for security risk analysis. p. 659–679. Available: http://www.sciencedirect.com/science/article/pii/S0167404810000209/pdf?md5=7c1137496ad99d8947ced7f4f032f05c&pid=1-s2.0-S0167404810000209-main.pdf

[2] H.-S. Rheea, C. Kim, and Y. U. Ryu. (2009, Self-efficacy in information security: Its influence on end users' information security practice behavior. p. 816–826. Available: http://www.sciencedirect.com/science/article/pii/S016740480900056X/pdf?md5=8d30439e1d493864f7650c02e95ec6d4&pid=1-s2.0-S016740480900056X-main.pdf

[3] J. Brito, A. Abreu, J. Bedoya, C. García, and C. Brizuela. (2013, Módulo diseñador de modelos para el generador dinámico de reportes. pp. 11. Available: http://publicaciones.uci.cu/index.php/SC/article/download/1258/711

[4] A. Rodriguez. (2010, Junio de 2010). Implementación del módulo Diseñador de Reportes para el Generador Dinámico de Reportes incluido en la Plataforma de Ayuda para la Toma de Decisiones y Soluciones Integrales. p. 80. Available: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_03391_10/1/TD_03391_10.pdf

[5] Y. Medinilla and M. Álvarez. (2013, Mayo de 2013). Herramienta para la migración de la base de datos del Generador Dinámico de Reportes (GDR) V1.8 a V2.0. [Tesis de pregrado]. p. 79. Available: http://bibliodoc.uci.cu/RDigitales/2013/diciembre/5/TD_07145_13.pdf

[6] J. Infante and Y. Hernández. (2009, Junio). Arquitectura de Software para el Sistema de Gestión de Reportes Dinámicos. p. 114. Available: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_2207_09/1/TD_2207_09.pdf

[7] A. Abreu, M. Lezcano, Y. Hernández, and A. Rodríguez. (2012, Generador Dinámico de Reportes versión 1.8.0. p. 13.

[8] CNSS, "National Information Assurance (IA) Glossary, CNSS Instruction No. 4009," C. o. N. S. Systems, Ed., ed, 2010, p. p. 103.

[9] R. Kanneganti and P. Chodavarapu. (2008). *SOA Security*.

[10] V. Suhendra. (2011, A Survey on Access Control Deployment. *259*, p. 11-20. Available: http://www1.i2r.a-star.edu.sg/~vsuhendra/2011-sectech.pdf

[11] Ueda and Ruggiero, "A Systematic Mapping on the Role-Permission Relationship in Role Based Access Control Models," ed, 2012, pp. p. 1243-1250.

[12] A. Vance, B. Molyneux, and P. B. Lowry, "Reducing Unauthorized Access by Insiders through User Interface Design: Making End Users Accountable " pp. 4623-4632, 2012.

[13] O. Gómez. (2012, Modelo de control de acceso para sistemas de información en entornos multidominios. p. 106.

[14] Y. Ma, X. Chen, L. Li, and Y. Luo. (2009, P2P-Based Single Sign-On. p. 845-846.

[15] P. Aubry, V. Mathieu, and J. Marchal. (2004, ESUP-Portail: open source Single Sign-On with CAS (Central Authentication Service). p. 9.

[16] Google. (2014, 15 de septiembre). *Cómo implementar SSO (inicio de sesión único)*. Available: http://support.google.com/a/answer/60224?hl=es&ref_topic=4388191

[17] M. Addison. (2010, 3 de septiembre). *CAS User Manual*. Available: http://wiki.jasig.org/display/CASUM/User+Manual

[18] E. Corporation. (2014). *CENTRAL AUTHENTICATION SERVICE (CAS) SSO FOR EMC DOCUMENTUM REST SERVICES*. Available: http://belgium.emc.com/collateral/white-papers/h12766-documentum-rest-cas-sso.pdf

[19] B. Donnelly and T. Poage. (2013). *About CAS*. Available: http://ucdavis.jira.com/wiki/display/IETP/About+CAS