PA1710-8-RSHIELD

RSHIELD: DISEÑO E IMPLEMENTACIÓN DE UN SERVICIO DE SEGURIDAD OPERANDO EN LA NUBE EN MODO MULTI-TENENCIA

CARLOS DANIEL PERDOMO FRANCO

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
2017

PA1710-8-RSHIELD
RSHIELD: DISEÑO E IMPLEMENTACIÓN DE UN SERVICIO DE SEGURIDAD OPERANDO EN LA NUBE EN MODO MULTI-TENENCIA

Autor:

Carlos Daniel Perdomo Franco

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE

MAGÍSTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Director

Ing. Lina María Consuelo Franky de Toro, PhD

Comité de Evaluación del Trabajo de Grado

Santiago Gil Bohórquez

Alexander Herrera Castro

Página web del Trabajo de Grado

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
Mayo, 2017

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS

Rector Magnífico

P. Jorge Humberto Peláez Piedrahita, S.J.

Decano Académico Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Decano del Medio Universitario Facultad de Ingeniería

Padre Antonio José Sarmiento Nova, S.J.

Directora Maestría en Ingeniería de Sistemas y Computación

Angela Cristina Carrillo Ramos

Director Departamento de Ingeniería de Sistemas

Efrain Ortiz Pabon

Artículo 23 de la Resolución No. 1 de Junio de 1946

"La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia"

AGRADECIMIENTOS

Agradezco a Dios por todas las bendiciones recibidas, por brindarme la salud y la fuerza que me permiten continuar y seguir creciendo cada día como persona y como profesional.

Agradezco infinitamente a mi directora de trabajo de grado, la Ingeniera y PhD Lina María Consuelo Franky de Toro quién con su conocimiento, apoyo y guía fue fundamental para el desarrollo de mi trabajo de grado, y la culminación exitosa de mi Maestría.

A mi familia, que han sido parte esencial en mi proceso ya que con su apoyo, consejos y ánimo me brindaron la fortaleza para no desfallecer en este proceso.

Contenido

Lista de figuras

Lista de tablas

ABSTRACT

Currently, the growth of the internet has directly impacted on the security of the information handled and exposed through this medium. This has driven the need for companies to implement security systems, especially in web applications. All of this has led companies to invest time and resources in the development and implementation of different security modules that allow them to protect the variety resources of software applications that are exposed to the web.

In the present work RSHIELD (Resource Shield) was designed and implemented. It is a multi-tenancy cloud-based security computing solution that allows multiple companies to perform a variety of role-based security configurations for different web applications. All this in a centralized place, without the need of being coupled to each application like the traditional solutions. This will allow RSHIELD's client companies to centrally control the different security configurations for their different web applications that use these services.

RESUMEN

En la actualidad, el crecimiento de internet ha impactado directamente en la seguridad de la información manejada y expuesta a través de este medio. Esto ha impulsado la necesidad de las compañías de implementar sistemas de seguridad especialmente en las aplicaciones web. Todo esto ha conllevado a las empresas a invertir tiempo y recursos en el desarrollo e implementación de diferentes módulos de seguridad que les permitan proteger los recursos de variedad de aplicativos de software expuestos a la web.

En el presente trabajo se diseñó e implementó RSHIELD, del inglés Resource Shield (Protector de Recursos). Es una solución informática de seguridad que opera en la nube en modo Multitenencia, que permite a múltiples compañías realizar variadas configuraciones de seguridad basadas en roles para diferentes aplicaciones web. Todo esto en un lugar centralizado, sin la necesidad de estar acoplado a cada aplicativo como sucede con las soluciones tradicionales de módulos de seguridad. Esto permitirá a las compañías clientes de **RSHIELD** tener de forma centralizada el control sobre las diferentes configuraciones de seguridad para sus diferentes aplicaciones web que hagan uso estos servicios.

RESUMEN EJECUTIVO

La difusión de las aplicaciones web en la actualidad ha despertado el interés por la sensibilidad de los datos expuestos a internet. De ahí la necesidad de implementar soluciones que permitan a las compañías proteger los recursos de sus aplicaciones y los datos accedidos a través de estos recursos. En su mayoría estas aplicaciones web expuestas por las compañías provienen de sistemas legados y/o sistemas que por necesidad y requerimientos de los clientes y la compañía han sido migrados como aplicaciones web o en algunos casos expuestos como servicios web. A su vez, muchos de estos sistemas y servicios están desarrollados en múltiples plataformas y lenguajes de programación, lo que ha conllevado a las compañías a desarrollar e implementar diferentes módulos de seguridad para proteger sus aplicaciones y servicios. Normalmente estos módulos desarrollados e implementados terminan prestando los servicios, con la única diferencia de que son implementados usando diferentes lenguajes de programación.

En el presente proyecto se aborda la necesidad de un sistema que ofrezca Servicios de Seguridad para la Identificación, Autenticación y Control de Acceso a múltiples compañías clientes y sus diferentes aplicaciones. Es así como en el presente trabajo de grado se diseña e implementa una solución informática que ofrece servicios de seguridad en la nube con un modelo de servicio SaaS que soporta Multi-tenencia y que a su vez permite de forma sencilla configurar la seguridad de sus aplicaciones basados en roles, los cuales posteriormente podrán ser accedidos por las empresas clientes a través de un conjunto de Servicios Web.

Con la solución informática RSHIELD producto del presente trabajo de grado se ofrece una solución centralizada que permite a múltiples tipos de empresas cliente acceder a una interfaz web en la cual puedan realizar distintas configuraciones de sus servicios de seguridad como: establecer el tipo de recursos a proteger, definir los diferentes roles, asignar los respectivos permisos sobre los recursos y asignar los usuarios o grupos de usuarios a los roles definidos previamente. Luego de realizadas las diferentes configuraciones de seguridad, las empresas cliente podrán acceder a estos servicios de seguridad y hacer uso de ellos en sus diferentes aplicaciones.

El presente trabajo de grado ofrece una alternativa a las soluciones típicas de seguridad eliminando el desgaste de tiempo y recursos que genera el desarrollo y/o acoplamiento de diferentes módulos de seguridad para cada tipo de aplicación. Ofrece un sistema centralizado de seguridad que brinda los servicios requeridos por las compañías para la protección de sus aplicaciones y recursos por medio del uso de un conjunto de servicios de seguridad.

INTRODUCCIÓN

El constante desarrollo tecnológico, el fácil acceso a la tecnología y la necesidad de mejorar procesos con el fin de brindar un valor agregado a los clientes han llevado a las compañías a adquirir productos de software y/o a desarrollar sus propias soluciones informáticas que se adapten a sus nuevas necesidades y requerimientos. En sus inicios estos sistemas fueron creados como sistemas cliente servidor, los cuales se encontraban dentro de las instalaciones de las compañías y podían ser accedidos por unos pocos equipos de cómputo. Con el avance tecnológico estos sistemas de información pasaron a ser sistemas alojados y accedidos a través de la web e internet y dejaron la red interna de las compañías. Sin embargo, las compañías siguieron desarrollando aplicaciones a la medida que colocaban en sus servidores para ser accedidos por los clientes a través de Internet. El siguiente paso consistió en dejar que las aplicaciones operaran en servidores externos a la compañía, administrados por terceros.

Es amplio el número de compañías que, por la necesidad de migrar a nuevas tecnologías, o a nuevos lenguajes, o por requerimientos del negocio han requerido desarrollar una variedad de sistemas de información. Todos estos sistemas de información tienen fines y funcionalidades diversas dependiendo del negocio y las necesidades específicas de cada compañía. Independiente de que sus líneas de negocio sean diferentes, todos estos sistemas tienen algo en común, pues la mayoría requiere hacer uso de servicios de seguridad para la Identificación, Autenticación y el Control de Acceso. Esto, con el fin de permitir y controlar el acceso a los diferentes servicios que prestan sus sistemas de información a sus diferentes usuarios.

Todas las compañías invierten tiempo y recursos en el desarrollo de módulos de seguridad para sus aplicativos o sistemas de información tanto internos como externos con el fin de realizar la Identificación, Autenticación y Control de Acceso (Autorización) [MART2016] para los usuarios de sus aplicaciones. Por lo general estos módulos en su mayoría están basados en seguridad por roles o Role Based Access Control (RBAC) [SAND1998]. Esto permite definir un conjunto de roles de acuerdo a las características de una aplicación, asociando un conjunto de permisos a cada rol y luego así asignar estos roles a los diferentes usuarios o grupos de usuarios.

El siguiente paso en la evolución de las aplicaciones informáticas de las compañías consistirá en dejar de implementar módulos de seguridad a la medida para acoplar a cada aplicación, y en lugar de ello empezar a utilizar soluciones informáticas de seguridad desarrolladas por terceros que prestan servicios a múltiples aplicaciones a la vez desde servidores en la Nube.

Con el fin de proveer servicios comunes o similares tanto para aplicaciones como para diferentes tipos de usuarios (compañías y/o personas), actualmente se han popularizado los servicios y sistemas de Cloud Computing [SOSI2011] o computación en la nube. El uso de la computación en la nube se ha masificado ya que ésta permite abstraer el detalle de la implementación de sus servicios a los usuarios. Es decir, que tanto la ubicación de almacenamiento del sistema y el lugar donde se encuentra ejecutándose son desconocidos por los usuarios de estos sistemas en la nube.

La computación en la nube o Cloud Computing tiene 3 modelos principales de prestación de servicios llamados modelos de prestación de nube ampliamente formalizados y aceptados. Estos son: Infraestructura como Servicio (IaaS), Plataforma como Servicio (PaaS) y Software como Servicio (SaaS) [HURW2012].

El modelo de software como Servicio (SaaS) es el modelo más maduro de los 3 modelos de servicios de la nube mencionados anteriormente. El modelo SaaS ofrece comúnmente 2 métodos para consumir o hacer uso de sus servicios. El más común, es a través de una interfaz web donde los usuarios acceden desde cualquier dispositivo conectado a internet. La segunda forma, es ofrecer un conjunto de APIs para que sus clientes (empresas) puedan consumir e integrar estos servicios ofrecidos dentro de sus propias aplicaciones [KAVI2014].

En el presente proyecto se acomete la necesidad de un sistema que ofrezca Servicios de Seguridad para la Identificación, Autenticación y Control de Acceso a múltiples compañías clientes y sus diferentes aplicaciones. Es así como el presente trabajo de grado diseña e implementa una solución informática que ofrece servicios de seguridad en la nube con un modelo de servicio SaaS que soporte Multi-tenencia y que a su vez permite de forma sencilla configurar la seguridad de sus aplicaciones basados en roles Role Based Access Control (RBAC), los cuales posteriormente podrán ser accedidos por las empresas clientes a través de un conjunto de Servicios Web [W3C2004].

Con el desarrollo del presente proyecto se desea ofrecer una solución informática centralizada denominada RSHIELD (del inglés Resource Shield que significa Protector de Recursos) que permita a diferentes tipos de empresas clientes acceder a una interfaz web en la cual puedan realizar distintas configuraciones de sus servicios de seguridad como: establecer el tipo de recursos a proteger, definir los diferentes roles, asignar los respectivos permisos a los roles y asignar los usuarios o grupos de usuarios a los roles definidos previamente. Luego de realizadas las diferentes configuraciones de seguridad las empresas clientes podrán acceder a estos servicios de seguridad y hacer uso de ellos en sus diferentes aplicaciones.

Como aclaración y por motivos de alcance del presente trabajo de grado, éste sólo contempla el diseño y desarrollo de la solución informática RSHIELD y el respectivo prototipo de validación. Por lo tanto, algunos requerimientos no funcionales como la escalabilidad y la alta disponibilidad no están contemplados en la presente entrega, a pesar de ser requerimientos que son típicamente bien resueltos por proveedores de computación en la nube.

Con el fin de evitar la exposición de información sensible de las compañías, en la solución informática desarrollada. RSHIELD solo manipulará dentro de su base de datos la información descriptiva de las aplicaciones registradas y la información de seguridad asociada a estas como roles, usuarios, URLs de aplicaciones entre otros.

I-PRESENTACIÓN DEL TRABAJO DE GRADO

En este capítulo, se describen los elementos principales que guiaron el desarrollo del presente trabajo de grado. Dentro de estos elementos se encuentran: los objetivos del proyecto, la metodología guía y el potencial de innovación de la solución propuesta.

1. Objetivos

De acuerdo a las problemáticas identificadas se plantearon para este proyecto un objetivo general y seis objetivos específicos; a continuación, se describe cada uno de ellos.

1.1. Objetivo General

Diseñar e implementar como solución informática un servicio Multi-tenencia de seguridad en la nube que permita a diferentes sistemas de información configurar y administrar el control de acceso basado en seguridad por roles y/o perfiles.

1.2. Objetivos Específicos

- ✓ Identificar alternativas tecnológicas para la implementación de la solución informática.
- ✓ Identificar y priorizar requerimientos para la solución informática.
- ✓ Diseñar la arquitectura de la solución informática a implementar.
- ✓ Desarrollar la solución informática que implemente los requerimientos priorizados.
- ✓ Diseñar y desarrollar un prototipo que permita probar la solución informática.
- ✓ Validar la solución informática

2. Metodología

El proceso de construcción de la solución informática "Diseño e implementación de un Servicio de Seguridad operando en la Nube en modo Multi-tenencia" se dividirá en las siguientes ocho fases:

1. Evaluación de soluciones similares.
2. Estudio y evaluación de alternativas tecnológicas.
3. Identificación y priorización de requerimientos.
4. Definición de funcionalidades.
5. Definición del diseño y arquitectura.
6. Desarrollo de la solución informática.
7. Diseño y Desarrollo del prototipo de la aplicación que prueba el Servicio de Seguridad.
8. Validación de la solución.

Dentro de las 4 primeras fases para el desarrollo de este proyecto se realizará toda la documentación, análisis y selección tanto de las mejores tecnologías para implementar la solución informática como de los requerimientos y funcionalidades prioritarias que serán implementadas. En la fase número 4 se pretende generar una visión general de los componentes de la solución y como estos se relacionarán e interactuarán entre ellos. Las fases 5 y 6 comprenden el desarrollo tanto del producto como de un prototipo que permita probar la solución diseñada e implementada en las fases anteriores. Por último, la fase 7 que comprenderá actividades de verificación y pruebas funcionales, con el fin de validar las funcionalidades de la solución implementada. Adicionalmente junto con la fase 8 se realizará una encuesta a posibles usuarios de la solución informática.

Fase 0. Estado del arte y análisis comparativo de soluciones de seguridad: Esta fase está contemplada para realizar un análisis del estado del arte de soluciones de seguridad existentes tanto de software libre como propietarias. Esto con el fin de realizar un análisis comparativo que permita evaluar bajo diferentes criterios las soluciones de seguridad existentes con respecto a la solución propuesta en el presente trabajo de grado.

Para el desarrollo de esta fase se realizará la siguiente actividad:

A. Análisis del estado del arte de soluciones de seguridad similares y comparativo con la solución propuesta.

Fase 1. Estudio y evaluación de alternativas tecnológicas: Esta fase consiste en realizar una documentación, análisis y finalmente una selección de las alternativas tecnológicas disponibles que serán utilizadas para el desarrollo e implementación de la solución informática propuesta. De esta misma forma se realizará una investigación de tecnologías para el manejo de datos en el contexto de Multi-tenencia.

Para esta etapa de selección de tecnologías se tienen como posibles opciones algunas tecnologías como: REST [FIELD2000] tecnología considerada para la implementación de los Servicios Web que permitirán a las empresas clientes acceder a los diferentes servicios de seguridad ofrecidos para la solución informática, Java Enterprise Edition (6/7) (JEE) [ORAC2014] como posible plataforma para el desarrollo tanto de los Servicios Web como de la solución informática, Cloud Computing, Software como Servicio (SaaS) como modelo de prestación de servicio en la nube, entre otras.

Para el desarrollo de esta fase se realizarán las siguientes actividades:

A. Investigación, análisis y documentación de alternativas tecnológicas.
B. Selección de tecnologías.
C. Documentación de las tecnologías seleccionadas.

Fase 2. Identificación y priorización de requerimientos: En esta fase de identificación y priorización de requerimientos serán considerados y valorados todos los posibles requerimientos que podrían ser incluidos en la solución informática a desarrollar. Semejante a la fase 1 esta fase estará apoyada de la metodología DAR con el fin realizar una correcta selección de los requerimientos que serán abarcados en la solución informática.

Como resultado de esta fase se obtendrá el conjunto de requerimientos más significativos y relevantes para ser incluidos en la solución informática.

Para el desarrollo de esta fase se realizarán las siguientes actividades:

A. Identificación y descripción de los requerimientos.
B. Definición de criterios de selección de requerimientos.
C. Selección de los requerimientos a incluir en la solución informática.
D. Documentación de los requerimientos seleccionados.

Fase 3. Definición de funcionalidades: Esta fase tiene como finalidad realizar la selección del conjunto de funcionalidades que serán implementadas en la solución informática a desarrollar. Estas funcionalidades estarán basadas en los requerimientos seleccionados y priorizados en la fase anterior de identificación y priorización de requerimientos.

Las funcionalidades seleccionadas en esta fase del proyecto serán gestionadas y documentadas como Casos de Uso [RUMB2004] los cuales estarán soportados por los requerimientos priorizados en la fase anterior.

Los casos de uso permiten describir el conjunto de secuencias e interacciones típicas entre los usuarios del sistema y el propio sistema y ofrece una perspectiva de cómo se utilizará el sistema [FOWL2003].

Para el desarrollo de esta fase se realizarán las siguientes actividades:

A. Descripción de los Casos de Uso identificados.
B. Definición de los criterios de selección de los Casos de Uso.
C. Selección de los Casos de Uso a implementar.
D. Documentación del inventario de Casos de Uso.

Fase 4. Definición del diseño y arquitectura: La solución propuesta al momento de ser utilizada por los usuarios contará con dos fases. Una fase de configuración ver Figura 1., y una fase de consumo de los servicios provistos por la solución informática, ver Figura 2.

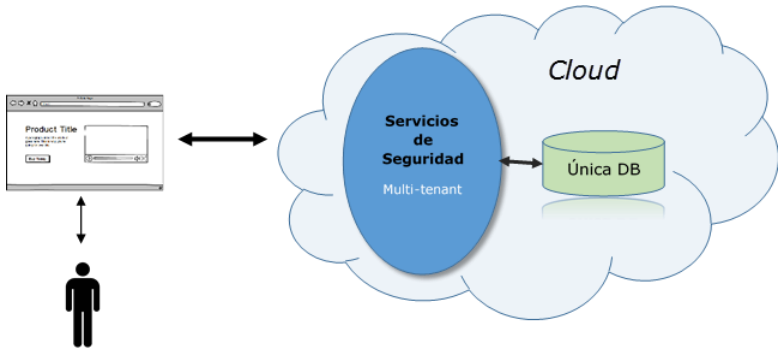

Figura 1. Diagrama de la fase de configuración de una aplicación cliente.

Como se puede observar en la Figura 1., la solución informática tendrá una interfaz web de administración a través de la cual una empresa cliente hará el registro de su aplicación, la configuración de usuarios y permisos para su posterior uso desde la aplicación de la empresa cliente.

En la Figura 2. Se muestra un diagrama general de la arquitectura deseada para la solución informática a desarrollar. Para la cual se requiere soporte Multi-tenencia o servir a diferentes tipos de empresas clientes y que sea una solución con un modelo de Software como Servicio del inglés SaaS.

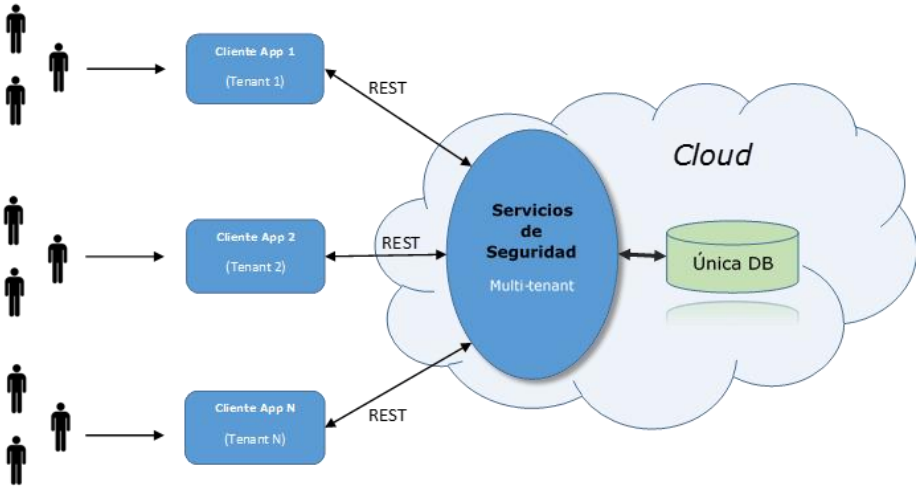

Figura 2. Visión General de La Arquitectura Solución Propuesta.

Los clientes directos para la solución informática desarrollada serán otras aplicaciones que requieran servicios de seguridad para proteger sus recursos por medio de Control de Acceso Basado en Roles o RBAC. El acceso a los servicios de seguridad de la solución informática será a través de servicios web REST a los cuales se les implementará un sistema que permita realizar una transmisión segura entre la aplicación cliente y los servicios de seguridad.

Para el desarrollo de esta fase se realizarán las siguientes actividades:

A. Definición de la arquitectura.
B. Diseño de la arquitectura.
C. Documentación del diseño y arquitectura.

Fase 5. Desarrollo de la solución informática: Esta fase está enfocada en desarrollar y obtener una versión funcional de la solución informática. Para esta etapa de desarrollo se hará uso de una metodología ágil de desarrollo de software *Extreme Programming* (XP) [BECK2004]. Esta metodología permitirá segmentar la etapa de desarrollo en un subconjunto de iteraciones de las que en cada una se obtendrá como salida una nueva versión incrementada y funcional de la solución informática con una nueva funcionalidad implementada.

Como se observa en la Figura 3. La fase de desarrollo se dividirá en un conjunto de iteraciones las cuales estarán precedidas de un plan de entregas o *Release Planning* [WELL2009], el cual será elaborado con base en el listado de casos de uso a implementar en la solución informática. Luego de establecido en plan de entregas se realizarán cada una de las respectivas iteraciones con las actividades detalladas en la Figura 3.

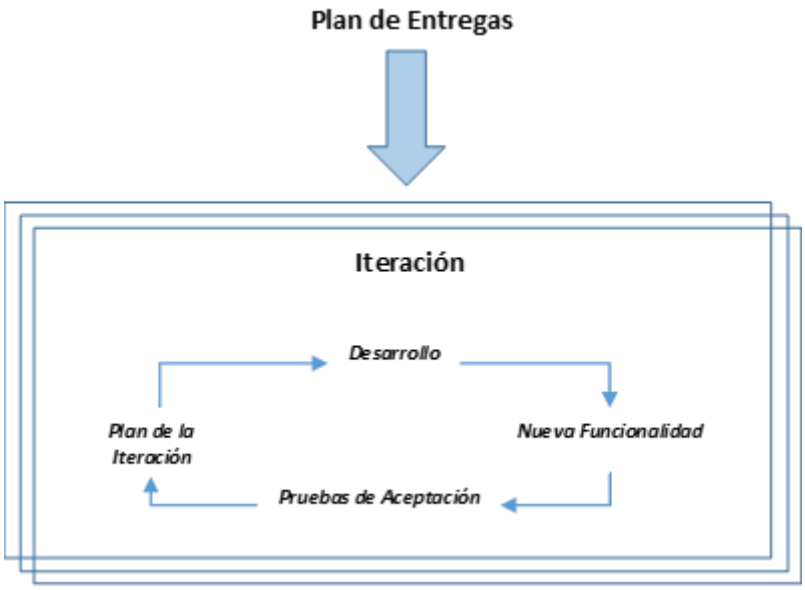

Figura 3. Ciclo de vida de las iteraciones a realizar para el proyecto.

Para el desarrollo de esta fase se realizarán las siguientes actividades:

A. Definición del plan de entregas (Definición del conjunto de iteraciones a realizar).
B. Actividades por Iteración:
    a. Definición de pruebas unitarias.
    b. Desarrollo de la solución.
    c. Implementación de pruebas unitarias.
    d. Pruebas de aceptación.
C. Documentación técnica del desarrollo.

Fase 6. Diseño y desarrollo del prototipo de la aplicación que prueba el servicio de seguridad: Como resultado de esta fase se pretende construir un producto de software prototipo que consuma y haga uso de los servicios ofrecidos por la solución informática desarrollada con el fin de mostrar cómo se realizaría la integración desde una aplicación cliente con la solución informática desarrollada y así validar el correcto funcionamiento de los servicios ofrecidos por el sistema desarrollado en este trabajo de grado.

El desarrollo del prototipo se realizará bajo la metodología XP en una iteración de entre 2 y 3 semanas, esto por restricciones de tiempo y cronograma.

Para el desarrollo de esta fase se realizarán las siguientes actividades:

A. Descripción del prototipo.
B. Diseño del prototipo.
C. Desarrollo del prototipo.
D. Validación del prototipo.

La finalidad de esta fase es demostrar de forma práctica cómo una aplicación cliente realiza la configuración de seguridad a través de la interfaz web expuesta por la solución desarrollada y luego mostrar como los usuarios de la aplicación cliente están regulados por el servicio de seguridad cuando usan las opciones ofrecidas por la aplicación cliente.

Fase 7. Validación de la solución: Esta fase consiste en diseñar y ejecutar un conjunto de pruebas funcionales que permitan probar que la solución informática desarrollada cumpla con las funcionalidades seleccionadas a ser implementadas. Adicionalmente se realizará una encuesta ante posibles usuarios de la solución.

Para el desarrollo de esta fase se realizarán las siguientes actividades:

A. Definición de los casos de prueba.
B. Ejecución del caso de prueba.
C. Evaluación de los resultaos obtenidos.
D. Documentación de los resultados obtenidos.
E. Realización de encuesta a posibles usuarios del servicio de seguridad.

3. Potencial de Innovación

Se realizó una investigación preliminar de productos de seguridad similares al construido en este proyecto, dentro de los cuales los más relevantes son: el servicio de autenticación de Google Google Identity Platform [GOOG2016] el cual ofrece servicios de autenticación haciendo uso de las cuentas de Gmail y Google +, y también están los servicios ofrecidos por OpenIAM [OPEN2016] de autenticación y autorización el cual es un producto que se puede descargar para instalar o comprar un servidor físico con estos módulos instalados. Además, de estas opciones existen algunas iniciativas de seguridad que consiste en módulos que deben ser acoplados a las aplicaciones que desean usar estos servicios.

En la fase 0 de este proyecto se revisarán múltiples soluciones de seguridad y se compararán con la solución propuesta desde el punto de vista de varios criterios, con el fin de justificar la solución propuesta.

Con el desarrollo de la solución informática “RSHIELD: Diseño e implementación de un Servicio de Seguridad operando en la Nube en modo Multi-tenencia” se pretende ofrecer no solo servicios de Autenticación sino de Autorización también; de igual forma al ser una solución implementada como modelo SaaS, no requerirá por parte de los usuarios ningún tipo de instalación del aplicativo. Esta solución permitirá la configuración de roles jerárquicos, restricciones

de tiempo sobre los recursos, tener un modelo de empresas con sucursales lo que permitirá estructurar la información correspondiente a la compañía dentro de la solución informática.

El producto del presente trabajo de grado será una solución de software libre lo que permitirá realizar modificaciones y adaptaciones según lo requiera cada cliente. Así mismo, la presente solución no requiere acoplarse a ninguna aplicación lo cual permitirá prestar los servicios de seguridad a múltiples aplicaciones cliente desarrolladas en diferentes plataformas y lenguajes de programación.

La implementación y uso del producto de este trabajo podría significar una oportunidad comercial para aquellas empresas que desean adaptar o extender la implementación resultado de este trabajo de grado. De igual forma que significaría un gran ahorro de tiempo y esfuerzos en desarrollo de módulos de seguridad para aquellos que deseen y hagan uso de los servicios provistos por la solución informática resultado de este trabajo de grado.

II-MARCO CONTEXTUAL

La necesidad de proteger los recursos y la información expuesta a internet ha sido el principal promotor para que las empresas busquen la forma de proteger los datos expuestos a través de sus sitios y aplicaciones web.

Una implementación tradicional de seguridad en aplicaciones web es hacer usos de los servicios ofrecidos por la plataforma de desarrollo y el lenguaje de programación utilizados, como es el caso de JEE [GUPT2013]. En aplicaciones desarrolladas haciendo uso de la plataforma JEE la seguridad de las aplicaciones es implementada haciendo uso de Java Authentication and Authorization Services (JAAS) [ORAC2014]. JAAS por su parte se puede realizar con dos propósitos: El primero para autenticación, que permite identificar quien está actualmente ejecutando la aplicación. Por otro lado, está la autorización, la cual permite asegurar que el usuario conectado posea los permisos adecuados para acceder a un recurso o recursos específicos.

Así como en el caso de JEE existe JAAS, para las aplicaciones desarrolladas tanto bajo la plataforma .NET [MICR2017], PHP [PHP2017] y otros lenguajes de programación, existe frameworks y métodos que permiten la implementación de seguridad para la autenticación y/o autorización que permiten proteger los recursos y los datos de las aplicaciones web.

Aún con los servicios de seguridad que las plataformas de desarrollo más populares ofrecen, en un entorno híbrido con aplicaciones desarrolladas en diferentes lenguajes de programación se dificulta la tarea de centralizar el control de seguridad de estas aplicaciones en un solo sitio. Lo cual conlleva a las compañías al desarrollo o adquisición de diferentes módulos que les brinden los servicios de seguridad para su variedad de aplicaciones.

Otro concepto que se ha popularizado actualmente es el concepto de Nube o Cloud. Actualmente son muchas las empresas que desean aprovechar este contexto, pero pueden no están preparadas, por lo cual podrían beneficiarse de los servicios de un tercero que ofrezca servicios en modo multi-tenencia especialmente servicios de seguridad.

III-MARCO TEÓRICO

A continuación, se explica brevemente las tecnologías, conceptos y demás elementos que sirvieron de apoyo, y que fueron utilizados para el desarrollo del presente proyecto.

1. Control de Acceso

El concepto de control de acceso tiene sus inicios en los años 70 con el surgimiento de los sistemas multi-usuario. Del cual nació la necesidad de restringir los recursos de los sistemas con el fin de asegurar que solo los usuarios autorizados tuvieran accesos a estos datos y recursos.

1.1. Fundamentos del control de acceso

El control de acceso es un concepto fundamental en una solución informática, el cual permite abordar los 3 más grandes riesgos de seguridad ampliamente conocidos y categorizados como lo son: la confidencialidad, la integridad y la disponibilidad. A continuación, se describe brevemente cada uno de ellos [FERRA2007]:

Confidencialidad: esta categoría hace referencia a la necesidad de mantener la información segura y privada. Un componente clave para proteger la confidencialidad de información es el **cifrado**. Cifrado es el proceso de codificación de un mensaje u otra información de modo que no se puede leer fácilmente [EAST2012, Capítulo 8]. El **cifrado** asegura que sólo las personas adecuadas es decir las personas que conocen la clave pueden leer la información. El uso de cifrado está muy extendido actualmente y se puede encontrar en casi todos los principales protocolos de comunicación en uso.

Integridad: Esta categoría hace referencia al concepto de proteger la información para que esta no sea modificada o alterada por usuarios no autorizados.

Disponibilidad: Esta categoría hace referencia al concepto en el cual la información debe estar disponible para usar cuando esta sea requerida.

El control de acceso es fundamental para preservar la confidencialidad e integridad de la información. Por un lado, la confidencialidad exige que sólo los usuarios autorizados puedan leer la información (datos, y/o recursos), y por su parte la integridad requiere que sólo los usuarios autorizados pueden alterar la información. El control de acceso es menos fundamental para la preservación de la disponibilidad, pero está claro que tiene un rol muy importante: Por Ejemplo, un atacante que obtenga acceso no autorizado a un sistema le será más fácil derribarlo [FERRA2007, Capítulo 1].

1.2. Autenticación y Autorización

Tanto la autenticación como la autorización hacen parte fundamental del control de acceso. Son dos conceptos que, aunque distintos son a menudo confundidos por su estrecha relación y dependencia el uno del otro, ya que una adecuada autorización es dependiente de la autenticación.

La **autenticación** [FERR2007, Capítulo 1] es el proceso de determinar que la identidad presentada por un usuario sea legítima. Normalmente los usuarios de computadores están familiarizados con las contraseñas, la cual es la forma más común de autenticación. Otras formas de autenticación, tal como la autenticación biométrica incluye, por ejemplo: lectores de huellas digitales y tarjetas inteligentes. La autenticación o el proceso de autenticación está basado en uno o más de los siguientes factores:

- Algo que sabes, como la contraseña, identificación personal número (PIN), o la combinación de la cerradura.
- Algo que tiene, por ejemplo, una tarjeta inteligente, cajero automático (ATM), tarjeta o llave.
- Algo que eres, o una característica física, como una huella dactilar o un patrón de retina, o una característica facial.

Claramente, la autenticación es normalmente más fuerte si se utilizan dos o más factores. Es decir, una contraseña puede ser adivinada, una clave puede perderse, y los sistemas de reconocimiento de rostros tienen una tasa de falsos positivos significativos, por lo que usar sólo uno de estos métodos o factores de autenticación puede no proporcionar un nivel aceptable de seguridad.

Mientras que la autenticación es un proceso para determinar quién eres, la **autorización** determina lo que le está permitido hacer o lo que puede hacer. La autorización se refiere a la decisión de si se le concede o no a un usuario el acceso a un recurso del sistema. Un sistema de información debe mantener una cierta relación entre el ID de usuario y los recursos del sistema, posiblemente, adjuntando una lista de usuarios autorizados a los recursos, o mediante el almacenamiento de una lista de recursos accesibles con cada ID de usuario. Hay que tener en cuenta que la autorización depende necesariamente de una correcta autenticación. Si el sistema no puede estar seguro de la identidad de un usuario, no hay manera de determinar si se le debe conceder acceso al usuario [FERR2007, Capítulo 1].

1.3. Usuarios, sujetos, objetos, operaciones, y permisos

Casi cualquier modelo de control de acceso puede establecerse usando formalmente las nociones de usuarios, sujetos, objetos, operaciones, permisos, y las relaciones entre estas entidades [FERR2007, Capítulo 1].

- El término **usuario** se refiere a las personas que interactúan con el sistema informático, y a una instancia de diálogo de un usuario con un sistema se denomina una sesión.
- Un proceso computacional que actúa en nombre de un usuario se denomina un **sujeto**.

- Un **objeto** se define como cualquier tipo de **recurso accesible** en un sistema informático. Incluyendo: archivos, periféricos como impresoras, bases de datos y entidades de granularidad fina tales como campos individuales de los registros de una base de datos. Los **objetos** son tradicionalmente vistos como entidades pasivas que contienen o reciben información, aunque los modelos de control de acceso incluyen la posibilidad de tratar los programas, impresoras, u otras entidades activas como objetos.
- Una **operación** es un proceso activo invocado por un sujeto.
- Los **Permisos** o privilegios son autorizaciones para llevar a cabo algún tipo de operación en el sistema. En la mayoría de la literatura el término permiso se refiere a una asociación de objeto y operación. Una operación particular usada en dos objetos diferentes representa dos permisos diferentes, y de manera similar, dos operaciones diferentes aplicados a un solo objeto representan dos permisos distintos.

2. Control de Acceso Basado en Roles RBAC

El Control de Acceso Basado en Roles RBAC (Role Based Access Control) [SAND1998] es conceptualmente simple: consiste en acceder a los objetos de un sistema informático basado en los roles que posee un usuario en la organización. Roles con diferentes privilegios y responsabilidades han sido reconocidos en las organizaciones, y en aplicaciones informáticas comerciales que se remontan a por lo menos la década de 1970, quienes ya implementaban limitadas formas de restricción de acceso basados en el rol del usuario dentro de la organización [FERR2007].

En la década de finales de 1980 y principios de 1990 los investigadores comenzaron a reconocer las virtudes de los roles como una abstracción para la gestión de privilegios dentro de las aplicaciones y sistemas de gestión de bases de datos. Un rol fue visto como un trabajo o posición dentro de una organización.

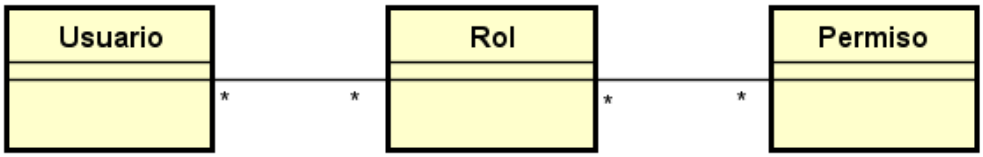

Figura 4. Relación RBAC.

Basados en la descripción hecha por Ferraiolo and Kuhn [FERR1992]. Un **rol** es esencialmente un conjunto de permisos. Por lo tanto, todos los usuarios reciben permisos sólo a través de los roles que les fueron asignados, como se muestra en la Figura 4. Dentro de una organización, los roles son relativamente estables, mientras que los usuarios y los permisos son numerosos y pueden cambiar rápidamente. Es así que, al controlar todos los accesos a través de roles, por tanto, simplifica la administración y revisión de los controles de acceso [FERR2007].

Una segunda característica de la descripción realizada por Ferraiolo and Kuhn [FERR1992] son los **roles jerárquicos**, (ver Figura 5.) donde los roles pueden heredar permisos de otros roles.

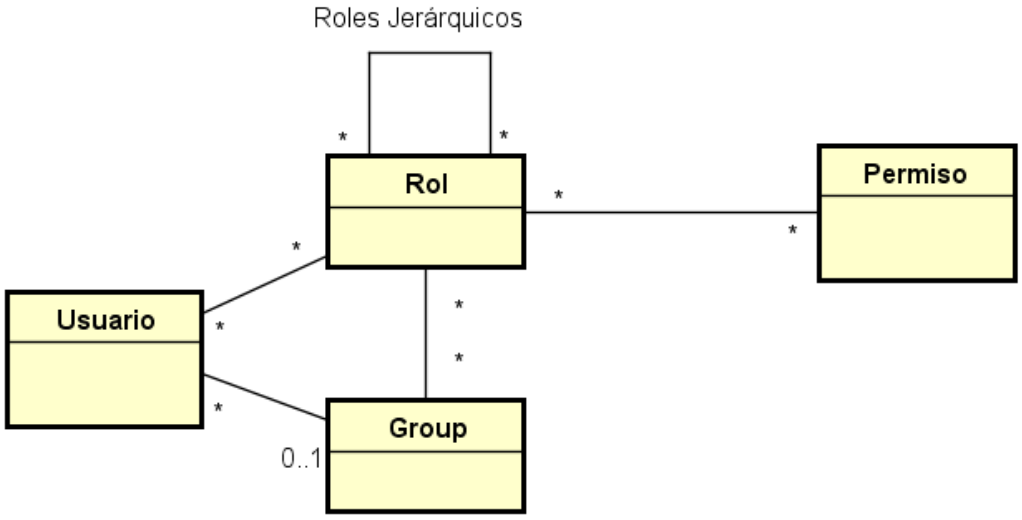

Figura 5. Jerarquías de Roles

Por otro lado, están los **grupos** (ver Figura 5), que son normalmente colecciones planas de usuarios. Esto permite que se puede asociar permisos tanto a grupos de usuarios como a usuarios individuales.

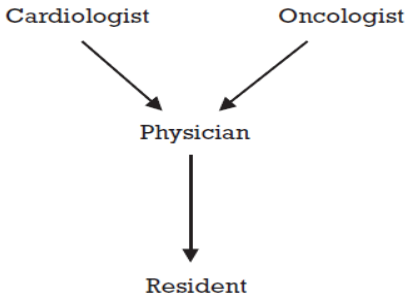

Figura 6. Ejemplo de una Jerarquía de Roles [FERR2007].

Como se puede observar en la figura 6. las jerarquías son un medio natural de estructurar roles para reflejar las líneas de autoridad y responsabilidades de una organización. Las jerarquías de Roles como se observa en la figura 6, permite heredar a otros Roles permisos previamente asignados a otros Roles. Es decir, en el momento que un Rol hereda otro hereda con este todos los permisos asociados al Rol del cual esté heredando.

Tal como se muestra en la figura 6, si se crea un nuevo Rol llamado (Cardiologist) o Cardiólogo ya que este es también un médico se le puede asignar a este el Rol previamente creado llamado (Physician) o médico y así el cardiólogo podrá también contar con los permisos asignados a un médico.

3. Soluciones de Seguridad

En el presente capítulo se relacionarán las herramientas más destacadas que ofrecen servicios de seguridad. Luego, en el capítulo 4.1 se realizará un comparativo entre estas herramientas y la solución de seguridad propuesta en el presente trabajo de grado.

3.1. OPENIAM

Identidad y Control de Acceso para las Empresas (*Identity and Access Management*): OpenIAM Access Manager es una solución tanto Open Source como licenciada, robusta, escalable, construida bajo una arquitectura orientada a servicios. Esta herramienta se integra perfectamente con otros productos como el Identity Manager para ofrecer una solución integral que permite tomar el control no sólo de quién puede acceder al sistema, sino también de lo que puede hacer una vez haya ingresado al sistema [OPEN2016].

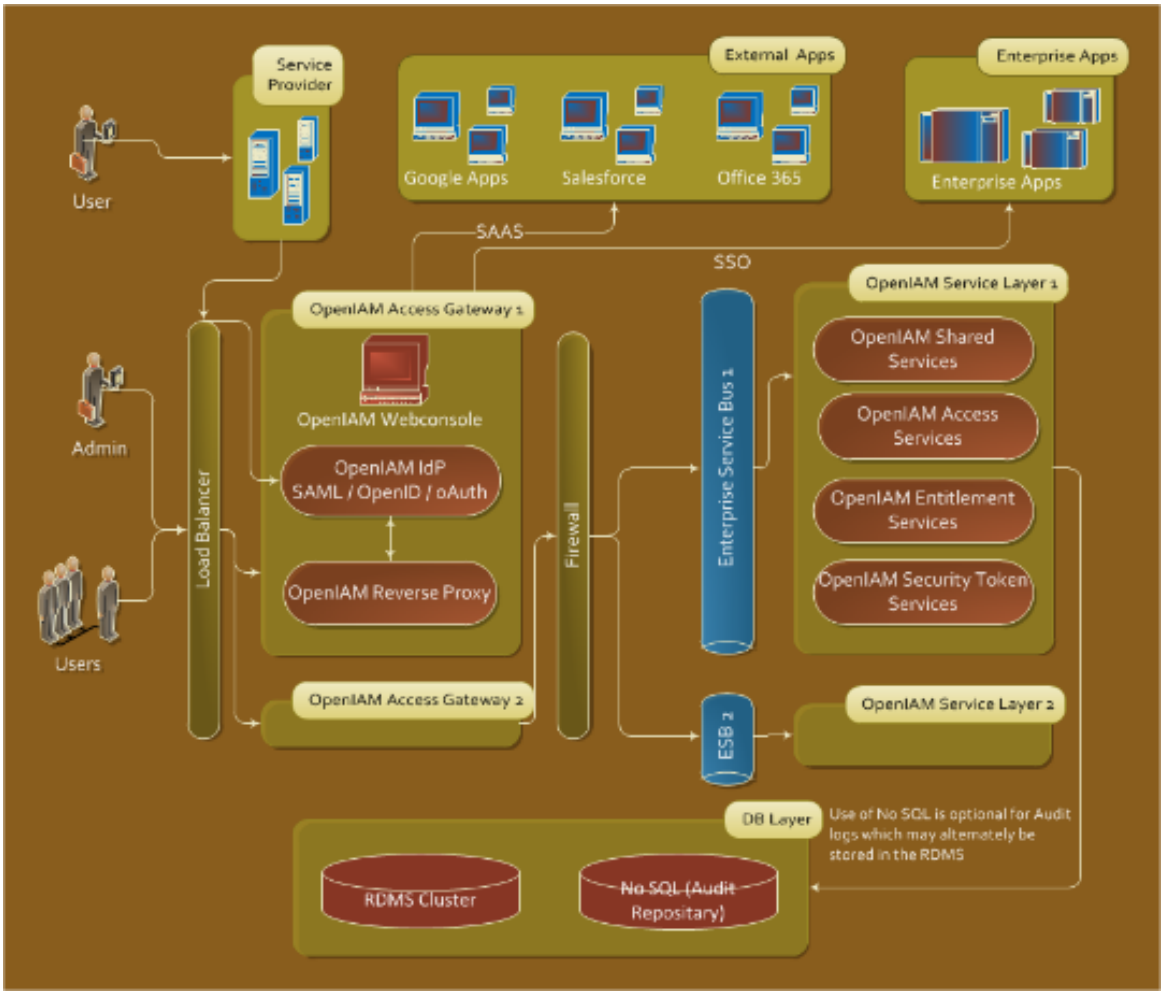

Figura 7. Visión general de la Arquitectura de OpenIAM

En la Figura 7 se puede observar la arquitectura general de la solución OpenIAM. A continuación, se listan las principales características de la solución OpenIAM.

Características Principales:

- Autenticación
- Autorización
- Gestión de Contraseñas
- Control de Acceso Basado en Roles RBAC
- XACML
- Federación & Single Sign-On (SSO[1])
- SOA Security.

OpenIAM hace uso de *eXtensible Access Control Markup Language* (XACML) [OASI2013] para la implementación de la autorización, lo cual le permite tener una forma flexible de definir reglas para el acceso a los recursos basadas en los usuarios, roles de los usuarios, el entorno, tiempo y fechas etc. Otro de los aspectos ofrecidos por OpenIAM es la gestión de contraseñas para los usuarios ofreciendo servicios como: gestión de históricos de contraseñas, definición de patrones de contraseñas entre otros.

En cuanto a gestión de usuarios OpenIAM ofrece el servicio de Federación de usuarios y Single Sign-On el cual permite administrar diferentes fuentes de usuarios como lo son LDAP, Directorios Activos y Bases de Datos relacionales.

3.2. WSO2

Servidor de Identidad (*WSO2 Identity Server*): Es un servidor de gestión de la identidad y autorización que facilita la seguridad durante la conexión y la gestión de múltiples identidades a través de diferentes aplicaciones [WSO22016].

Características Principales:

- Autenticación Fuerte
- Single Sign-On (SSO)
- Administración y Gestión de Usuarios y Grupos
- Control de Acceso Basado en Roles (RBAC)
- XACML

[1] SSO: Single Sign-On o inicio de sesión único es el mecanismo por el cual una sola acción de la autenticación y autorización de usuarios puede permitir a un usuario acceder a todos los equipos y sistemas en los que tiene permiso de acceso, sin la necesidad de introducir múltiples contraseñas. SSO reduce el error humano, un componente principal de fallo de los sistemas y, por tanto, es altamente deseable, pero en algunos casos difícil de implementar.

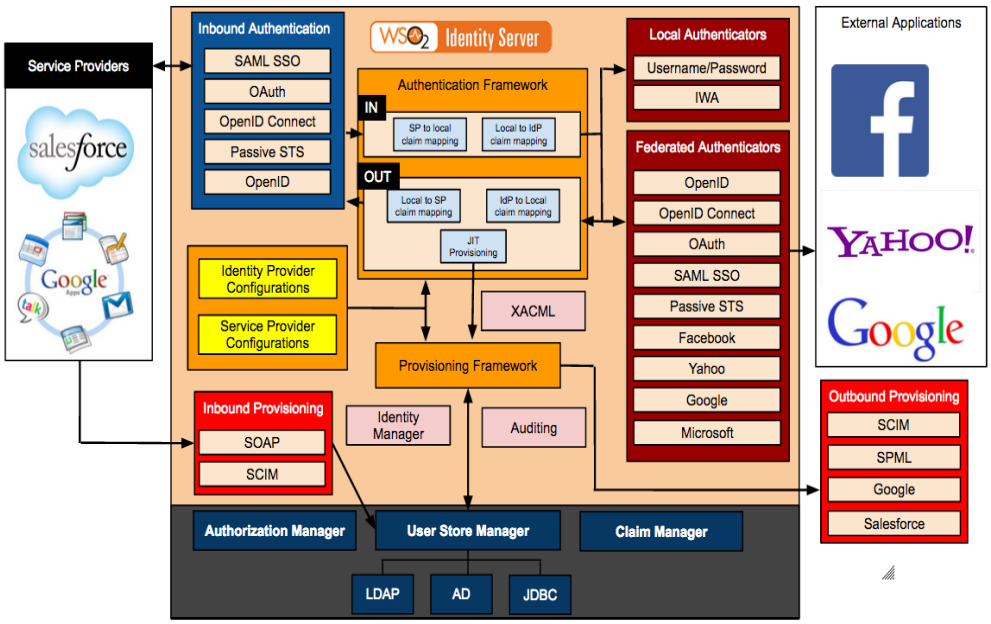

Figura 8. Arquitectura WSO2 Identity Server

El Identity Server ofrecido por WSO2 viene con un conjunto de herramientas como lo es la autenticación Fuerte. Esta autenticación Fuerte permite realizar autenticación utilizando los tres factores: Algo que usted sabe, Algo que usted posee, Algo que la persona es.

Otra herramienta ofrecida por WSO2 es la opción de integrase con múltiples aplicaciones externas y permitir a los usuarios autenticarse a través de estas aplicaciones.

La herramienta Identity Server ofrecida por WSO2 al igual que OpenIAM ofrece la posibilidad de tener una federación de usuarios a través de la integración con LDAP, Directorio Activo y Base de Datos relacionales.

3.3. FORGEROCK

Gestión de Accesos *(ForgeRock Access Management)*: "El señor de las cosas"; todo en una solución para la entrega de un acceso seguro en todo momento a los clientes, aplicaciones, dispositivos y cosas [FORG2016]. ForgeRock está basado en OpenAM. OpenAM es un servidor de gestión de acceso centralizado de código abierto, asegura los recursos protegidos a través de la red y proporciona autenticación, autorización, seguridad Web y servicios de federación en una única solución integrada. OpenAM gestiona el acceso a los recursos protegidos mediante el control de quién tiene acceso, cuándo, durante cuánto tiempo y bajo qué condiciones.

Características Principales:

- Autenticación
- Autorización

- Adaptive Risk Authentication
- Federation
- Single Sign-On (SSO)
- Auto Gestión de Usuarios

En figura 9, se puede observar la arquitectura de la solución de ForgeRock la cual está basada en OpenAM.

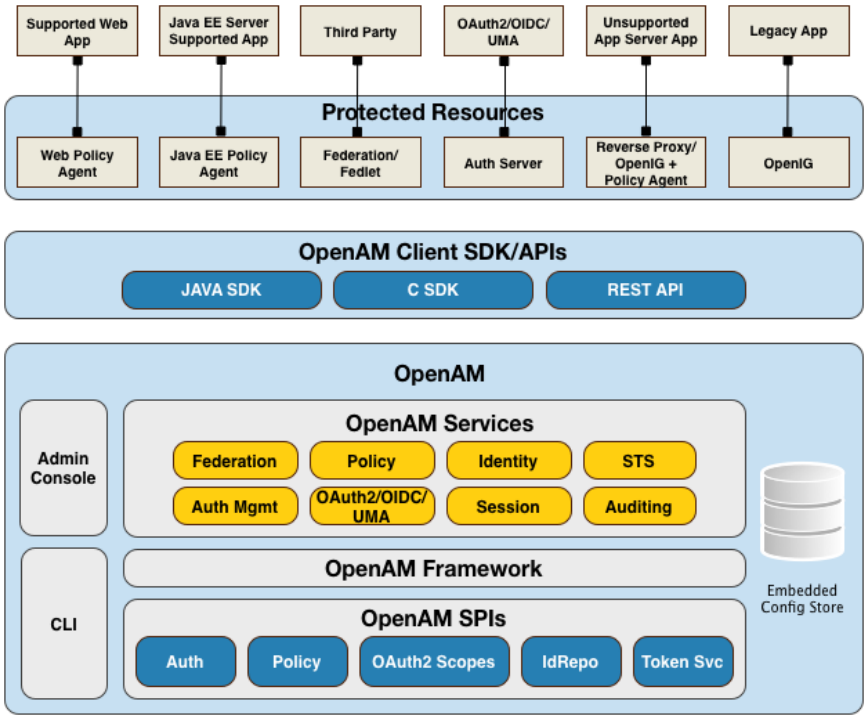

Figura 9. Arquitectura de ForgeRock Access Management

Dentro de las características a destacar de la herramienta ForgeRock está el variado conjunto de Apis provistas para integrase con esta solución. Dentro del conjunto de Apis ofrecido por ForgeRock está: Un kit de desarrolla para aplicaciones en Java Empresarial, un kit de herramientas de desarrollo para clientes desarrollados en lenguaje de programación C y finalmente un conjunto de servicios web REST con el Api para integrarse a esta solución.

3.4. ORACLE

Gestión de Accesos (*Oracle Access Management*): Oracle Access Management es una solución completa diseñada para permitir de forma segura la transformación del negocio con las tecnologías móviles y redes sociales, híbrido en las instalaciones y despliegue de aplicaciones de nube, y el despliegue de gestión de acceso híbrido preservando al mismo tiempo una experiencia de usuario sin fisuras, administración centralizada, y líder en el mercado en rendimiento y escalabilidad [ORAC2016].

Características Principales:

- Múltiples Esquemas de Autenticación
- Single Sign-On (SSO)
- Administración del ciclo de Vida de la Sesión
- Autenticación Gruesa

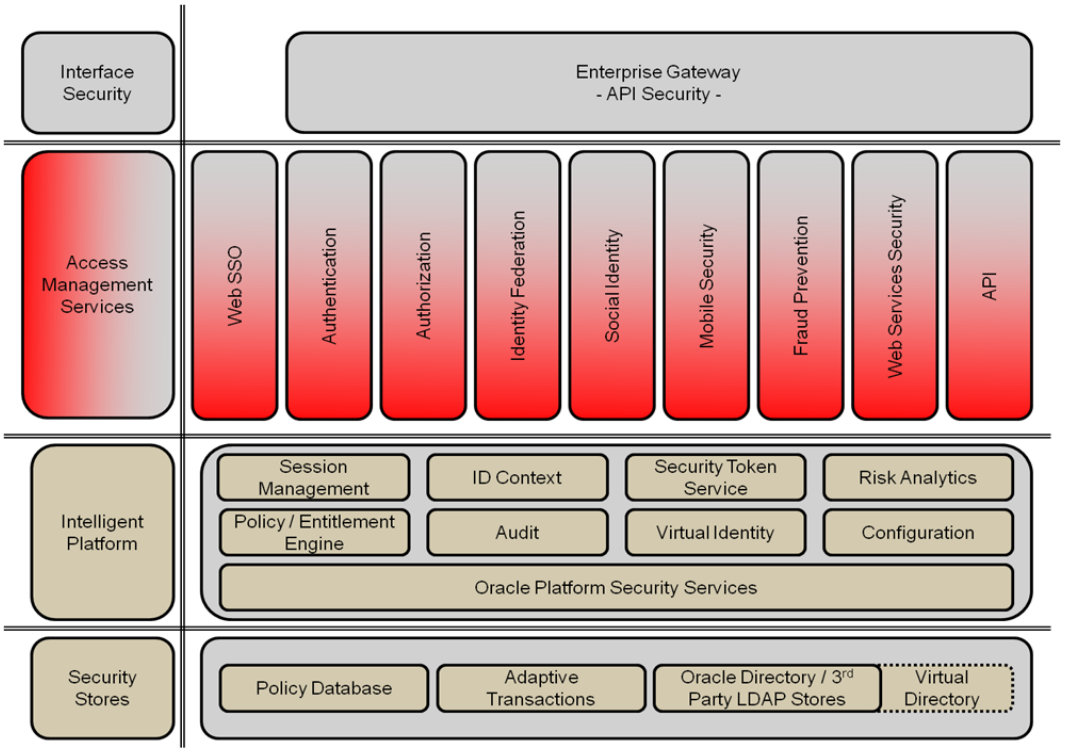

Figura 10. Visión general de la arquitectura de Oracle Access Management [ORAC2016]

La herramienta ofrecida por Oracle contiene un gran conjunto de servicios de seguridad para aplicaciones. El inconveniente con esta herramienta en que es licenciada con unos costos elevados para acceder a estos servicios.

Dentro de los elementos a resaltar de la herramienta provista por Oracle está el componente de detección de fraude que ofrece un nivel más inteligente de detención de fraude en tiempo real.

4. Computación en la Nube

La computación en la nube o *Cloud Computing* hace referencia tanto aplicaciones como servicios que corren sobre una red distribuida haciendo uso de recursos virtualizados que son accedidos a través de un protocolo de internet y de red estándar. La computación en la nube se fundamenta en la idea que los recursos son virtuales y sin límites y que los detalles de los sistemas físicos en los que se ejecuta el software o servicios se abstraen de usuario [SOSI2011].

La computación en la nube toma toda la tecnología, los servicios y aplicaciones que son similares a las de Internet y los convierte en una herramienta de auto-servicio. El uso de la palabra "nube" (*cloud*) hace referencia a dos conceptos esenciales:

- **Abstracción:** La computación en nube abstrae los detalles de la implementación del sistema de los usuarios y desarrolladores. Las aplicaciones se ejecutan en los sistemas físicos que no están especificados, los datos se almacenan en lugares que son desconocidos, la administración de sistemas se subcontrata a otros, y el acceso de los usuarios es ubicuo.
- **Virtualización:** La computación en nube virtualiza los sistemas poniendo en común los recursos y compartiéndolos. Tanto los sistemas como el almacenamiento pueden ser proporcionados desde una infraestructura centralizada cuando esta sea requerida, se habilita el soporte multiempresa o multi-tenencia [AGRA2011], y los recursos son escalables con facilidad y agilidad.

En el proceso de querer describir la computación en la nube se han definido dos tipos de *cloud* o nube. Estos tipos de nube son: aquellas basadas en el modelo de despliegue (*deployment model*) y aquellas basadas en el modelo de prestación de servicios (*service model*).

Al igual que se establecieron los modelos de despliegue y de implementación, la computación en la nube establece cinco características esenciales que los sistemas de computación en la nube deben ofrecer [SOSI2011]:

1. **Auto-servicio bajo demanda:** un cliente puede disponer de recursos informáticos sin la necesidad de interacción con el personal del proveedor de servicios *cloud*.

2. **Amplio acceso a la red:** El acceso a los recursos en la nube está disponible en la red mediante métodos estándar de una manera que permite el acceso independiente de la plataforma para todo tipo de clientes. Esto incluye una mezcla de sistemas operativos heterogéneos y plataformas gruesos y delgados, tales como ordenadores portátiles, teléfonos móviles, y PDA.

3. **Puesta en común de recursos:** Un proveedor de servicios *cloud* crea recursos que se agrupan juntos en un sistema que admite el uso de varios inquilinos. Sistemas físicos y virtuales se asignan o reasignan dinámicamente según sea necesario. En este concepto de puesta en común se encuentra intrínseca la idea de abstracción que oculta la ubicación de los recursos, tales como máquinas virtuales, procesamiento, memoria, almacenamiento y ancho de banda de red y conectividad.

4. **Elasticidad rápida:** Los recursos se pueden proveer rápidamente y elásticamente. El sistema puede añadir recursos por cualquiera de los sistemas de escalabilidad (ordenadores más potentes) u horizontal (más ordenadores del mismo tipo), y la escalabilidad puede ser automática o manual. Desde el punto de vista del cliente, los recursos de computación en la nube deben ser ilimitados y se pueden comprar en cualquier momento y en cualquier cantidad.

5. **Servicio Medido:** El uso de los recursos del sistema de nubes es medido, auditado y comunicado al cliente basado en un sistema de medida.

En la siguiente imagen se puede observar el conjunto de elementos que hacen parte de la definición de *cloud.*

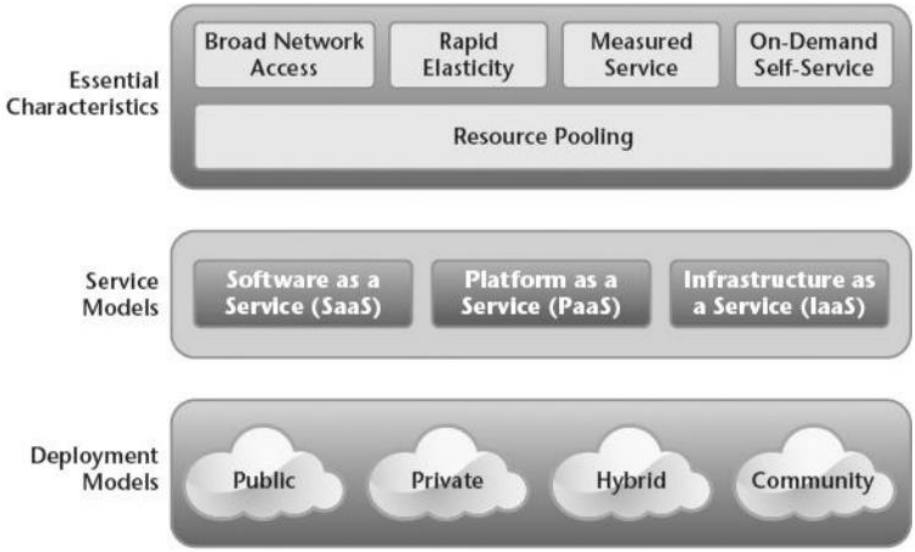

Figura 11. Definición de computación en la nube [SOSI2011].

4.1. Modelos de Despliegue

Un modelo de despliegue define el propósito de la nube y la naturaleza de cómo se encuentra la nube.

El Instituto Nacional de Estándares y Tecnología (NIST)[2] de los Estados Unidos define cuatro modelos de despliegue [MELL2009]. Los cuales, se describen a continuación:

- **Nube Pública:** La infraestructura de nube pública está disponible para uso público como alternativa para un gran grupo de la industria y es propiedad de una organización que vende servicios en la nube.

- **Nube Privada:** La infraestructura de nube privada es operado para el uso exclusivo de una organización. La nube puede ser gestionada por la organización o por un tercero. Las nubes privadas pueden estar tanto dentro como fuera de las instalaciones de la organización.

[2] NIST National Institute of Standards and Technology: https://www.nist.gov/

- **Nube Híbrida:** Una nube híbrida combina múltiples nubes (privada, comunitaria y pública), donde esas nubes retienen sus identidades únicas, sino que están unidas entre sí como una unidad. Una nube híbrida puede ofrecer acceso normalizado y propiedad de los datos y aplicaciones, así como la portabilidad de aplicaciones.
- **Nube Comunitaria:** Una nube comunitaria es aquella donde la nube se ha organizado para servir a una función o propósito común.

4.2. Modelos de Servicio

A medida que la computación en nube se ha desarrollado, diferentes proveedores ofrecen nubes que tienen diferentes servicios asociados a ellas. La cartera de servicios ofrecidos añade otro conjunto de definiciones llamados el modelo de servicio.

Hay muchos diferentes modelos de servicio descritos en la literatura, todos los cuales tienen la siguiente forma:

$$\text{XaaS, or "} < Something > \text{ as a Service"}$$

Dentro del gran número de modelos de servicio definidos existen tres que son ampliamente aceptados por la comunidad [SOSI2011, Capítulo 1]. Estos son:

- **Infraestructura como Servicio (IaaS):** IaaS ofrece máquinas virtuales, almacenamiento virtual, infraestructura virtual, y otros activos de hardware como recursos que los clientes pueden aprovisionar. El proveedor de servicios IaaS gestiona toda la infraestructura, mientras que el cliente es responsable de todos los demás los aspectos del despliegue. Esto puede incluir el sistema operativo, las aplicaciones y las interacciones del usuario con el sistema.
- **Plataforma como servicio (PaaS):** PaaS ofrece máquinas virtuales, sistemas operativos, aplicaciones, servicios, marcos de desarrollo, transacciones y las estructuras de control. El cliente puede desplegar sus aplicaciones en la infraestructura de la nube o hacer uso de aplicaciones que se programaron utilizando lenguajes y herramientas que son soportados por el proveedor de servicios PaaS. El proveedor de servicios gestiona la infraestructura de nube, los sistemas operativos entre otros. El cliente es responsable de la instalación y gestión de la aplicación que se está desplegando.
- **Software como Servicio (SaaS):** SaaS es un entorno operativo completo con aplicaciones, administración, e interfaz de usuario.
  En el modelo SaaS, la aplicación se proporciona al cliente a través de una interfaz de cliente ligero (un navegador, por lo general), y la responsabilidad del cliente comienza y termina con la entrada y la gestión de sus datos y la interacción del usuario. Todo, desde la aplicación hasta la infraestructura es responsabilidad del proveedor.

5. Multi-tenencia

Multi-tenencia o *(Multi-tenant*) es una de las características clave de SaaS o del modelo de servicios de Software como Servicio. La cual consiste en el concepto en el que una solo instancia de una aplicación que se encuentra alojada en un servidor puede servir o prestar el servicio a múltiples clientes u organizaciones que son llamados (*tenant*) como se puede observar en la Figura 9. [AGRA2011].

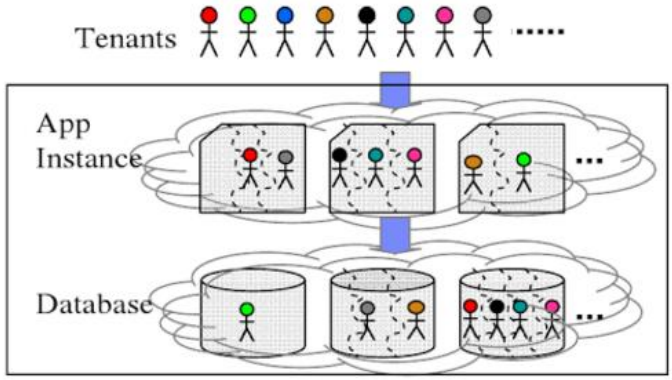

Figura 12. Entorno Multi-tenencia

El enfoque multi-tenencia puede traer una serie de beneficios que incluye el margen de beneficio mejorado para el proveedor de servicio a través de la reducción de costos de entrega y la disminución de costo de suscripción de servicios para los clientes. Esto hace que la oferta de servicios que ofrece sea atractiva para clientes potenciales, especialmente los clientes de pequeñas y medianas empresas en las cuales el presupuesto de inversión en TI es muy limitado [AGRA2011].

En un entorno multi-tenencia es indispensable un adecuado manejo y gestión de los datos o información de cada cliente, organización o tenant. Existen tres enfoques para la gestión de los datos en entornos multi-tenencia o patrones de uso compartido de recursos (*Resource Sharing Patterns)* [AGRA2011] los cuales se describen a continuación.

5.1. Totalmente Aislado (*Tatally Isolated*)

En este enfoque cada cliente o *tenant* posee una base de datos separada como se puede observar en la Figura 13.

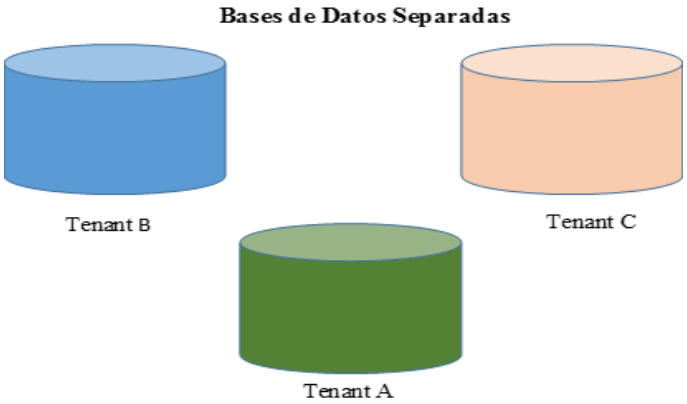

Figura 13. Enfoque de Bases de Datos Separadas.

5.2. Parcialmente Compartido (*Partially Shared*)

En este enfoque múltiples clientes o tenant comparten la misma base de datos, pero cada tenant posee su propio conjunto de tablas y esquema como se observa en la Figura 14.

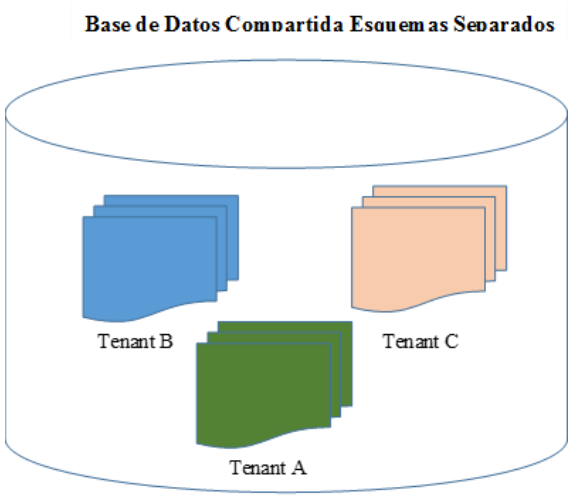

Figura 14. Enfoque de Bases de Datos Compartida con Esquemas Separados.

5.3. Totalmente Compartido (*Totally Shared*)

En este enfoque múltiples tenant comparten la misma base de datos, tablas y esquema. En este patrón todos los registros de todos los tenant son almacenados en una única tabla compartida, en donde en cada tabla se inserta una columna con el id del tenant la cual es utilizada para asociar cada registro con el tenant correspondiente como se observa en la Figura 15.

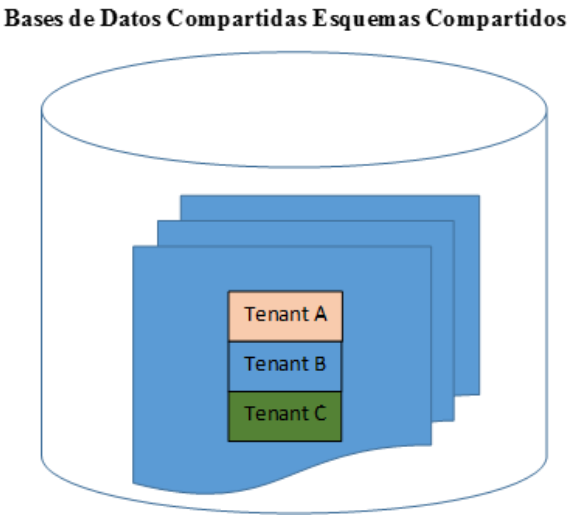

Figura 15. Enfoque de Bases de Datos Compartidas y Esquemas Compartidos.

Como se puede observar entre los tres patrones el patrón de bases de datos compartidas y esquemas compartidos parece ser el más eficiente entre los dos en cuanto al consumo de recursos e infraestructura. Sin embargo, al mantener todos los tenant en la misma base de datos y esquema acarrea otros inconvenientes en cuanto a la administración y mantenimientos de estas bases de datos, como por ejemplo el respaldo y restauración de backup de los datos de cada cliente.

6. Metodologías ágiles

En los años 90 comenzaron a surgir movimientos que se identificaban con el nombre de Metodologías Livianas o (*Lightweight Methodólogies*) los cuales se desarrollaron como reacción a las predominantes metodologías de peso pesado o (*Heavyweight Methodólogies*) de la época. Estas metodologías pesadas eran criticadas principalmente por ser rígidas y estar dominadas por una excesiva documentación [ALAI2013, Capítulo 1].

En el año 2001, se reunió un grupo de diecisiete personas referentes de las metodologías livianas y del desarrollo del software en Utah (Estados Unidos) con el fin de encontrar una alternativa a la problemática de los procesos de desarrollo de software tradicionales. De esta reunión nació la *Agile Alliance*[3], una organización sin ánimo de lucro encargada de promover la filosofía ágil y brindar apoyo a otras organizaciones en la adopción de esta filosofía. Así mismo, y como

[3] https://www.agilealliance.org/

resultado de esta reunión también nació el Manifiesto Ágil o (*Agile Manifesto*[4]) [ALAI2013, Capítulo 1].

El Manifiesto Ágil resalta cuatro valores los cuales son [BECK2016]:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Dentro de las metodologías agiles más representativas se encuentran: Programación Extrema o *Extreme Programming* (XP) y Scrum. En los siguientes capítulos se describirán las características de cada una.

6.1. Programación Extrema (XP)

Programación Extrema o *Extreme Programming* (XP) [BECK2005] es un estilo de desarrollo de software centrado en la correcta aplicación de técnicas de programación, comunicación clara, y en el trabajo en equipo. XP incluye:

- Una filosofía de desarrollo de software basado en los valores de la comunicación, la retroalimentación, la sencillez, el valor y el respeto.
- Un conjunto de prácticas probadas útiles para mejorar el desarrollo de software. Estas prácticas se complementan entre sí, amplificando sus efectos.
- Un conjunto de principios complementarios, técnicas para traducir los valores en práctica.
- Una comunidad que comparte estos valores y muchas de las mismas prácticas.

XP es una ruta de mejora hacia la excelencia la cual se distingue de otras metodologías por las siguientes características:

- Sus ciclos de desarrollo son cortos, permitiendo una retroalimentación temprana y continua.
- Su capacidad de programar de forma flexible la aplicación de la funcionalidad, responder a las necesidades cambiantes del negocio.
- Su confianza en las pruebas automatizadas escritas por los programadores, clientes y probadores para monitorear el progreso del desarrollo, para permitir que el sistema evolucione, y para detectar defectos de forma temprana.
- Confía en la comunicación verbal, las pruebas, y el código fuente para comunicar la estructura del sistema y su propósito.
- Confía en un proceso de diseño evolutivo que dura el tiempo que dura el sistema.

[4] http://agilemanifesto.org/

XP hace énfasis en el trabajo en equipo. Gerentes, clientes y desarrolladores son todos socios iguales en un equipo de colaboración. XP habilita un entorno sencillo, pero eficaz que permite a los equipos ser altamente productivos. El equipo se auto organiza alrededor del problema para resolverlo de la forma más eficientemente posible [WELL2009].

XP permite mejorar un proyecto de software en cinco formas esenciales; la comunicación, la sencillez, la retroalimentación, el respeto y el valor. Los programadores constantemente se comunican con sus clientes y colegas programadores. Mantienen su diseño simple y limpio. XP favorece el obtener retroalimentación ya que se realizan pruebas del software desde etapas tempranas del desarrollo.

6.2. Scrum

Scrum es una de las metodologías ágiles más populares. Es una metodología adaptativa, iterativa, rápida, flexible y eficaz, diseñada para ofrecer un valor significativo de forma rápida y en todo el proyecto. Scrum garantiza la transparencia en la comunicación y permite desarrollar un ambiente de responsabilidad colectiva y de progreso continuo [SCRU2016].

Scrum es una metodología que puede ser implementada en cualquier tipo de proyecto. Scrum es ideal para proyectos que cambian rápidamente. Scrum no define una forma detallada de cómo deben realizarse las tareas de un proyecto, este genera un contexto relacional e iterativo, de inspección y adaptación constante para que los involucrados en el proyecto vayan creando su propio proceso [ALAI2013].

Scrum se apoya en tres roles: *Product Owner*, *Development Team* y el *Scrum Master*. El Product Owner es quien representa al negocio, *stakeholders*, cliente y usuarios finales. El *Scrum-Master* es quien vela por la utilización de Scrum, la eliminación de impedimentos y asiste al equipo a que logre su mayor nivel de rendimiento posible. Y por último el *Development Team* o equipo de desarrollo que está compuesto por todo el personal necesario para la construcción del producto [ALAI2013]. Los roles y la relación entre ellos se pueden observar en la Figura 16.

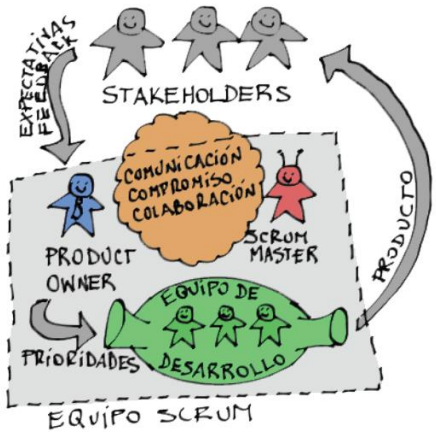

Figura 16. Scrum roles y sus relaciones [ALAI2013].

De forma resumida, en Scrum un proyecto se ejecuta por iteraciones de entre dos y cuatro semanas llamadas *Sprints*. Al inicio de cada *Sprint* el equipo se compromete a desarrollar un conjunto de nuevas funcionalidades o características del producto y al finalizar dicho Sprint se espera que dichas funcionalidades estén terminadas. En este momento es donde se realizar una reunión con el *Product Owner* con el fin de mostrar la funcionalidad desarrollada y obtener retroalimentación.

7. Tecnologías, herramientas y Frameworks de Desarrollo

A continuación, se mencionan las distintas tecnologías que sirvieron de apoyo para el desarrollo de la solución informática R-SHIELD.

7.1. Java Enterprise Edition

Java Enterprise Edition (JEE) [GUPT2013] es una plataforma para el desarrollo de aplicaciones web y empresariales, Las aplicaciones empresariales normalmente son diseñadas como aplicaciones multinivel. Las aplicaciones multinivel más comunes están compuestas por tres niveles, un nivel de interfaz gráfica de usuario apoyada en frameworks de desarrollo web que provee la plataforma, un nivel intermedio que provee servicios de seguridad y transacciones, y un nivel que provee los servicios de conectividad a bases de datos y sistemas legados.

La Plataforma JEE define un conjunto de APIs para los diferentes componentes que se encuentran en cada uno de los niveles. Estos componentes desarrollados con estas APIs provistas por la plataforma pueden ser desplegados en un contenedor que provea soporte de ejecución. Estos componentes una vez desplegados podrán comunicarse con otros componentes a través de protocolos y métodos ofrecidos por el contenedor.

JEE ha evolucionado con el tiempo y actualmente se encuentra en su versión 7. Java EE 7 fue liberado en junio del 2013 y provee un simple, fácil de usar y completo pilar herramientas para el desarrollo de aplicaciones web y empresariales.

7.2. Apache Maven

Apache Maven [MAVE2017] es una herramienta para la gestión y construcción de proyectos Java. Está basado en el concepto de un modelo de objeto de proyecto o *Proyect Object Model* (POM), Maven puede gestionar la construcción, la gestión de dependencias, la generación de informes y documentación de un proyecto a partir de una pieza central de información.

El principal objetivo de Apache Maven es permitir a un desarrollador comprender el estado completo de un esfuerzo de desarrollo en el menor período de tiempo. Para alcanzar este objetivo hay varias áreas de preocupación que Maven intenta abordar:

- Hacer que el proceso de construcción sea fácil
- Proporcionar un sistema de construcción uniforme

- Proporcionar información sobre la calidad del proyecto
- Proporcionar directrices sobre mejores prácticas de desarrollo
- Permitir la migración transparente a nuevas características

Algunas de las características más representativas que ofrece Apache Maven son:

- Configuración sencilla del proyecto que sigue las mejores prácticas (Obtener un nuevo proyecto o módulo funciona en segundos).
- Capaz de trabajar fácilmente con varios proyectos al mismo tiempo.
- Extensible, con la capacidad para escribir fácilmente plugins en Java o lenguajes de scripting.
- Acceso instantáneo a nuevas características con poca o ninguna configuración adicional.
- Construcción Basada en Modelos: Maven es capaz de construir cualquier número de proyectos en los tipos de salida predefinidos, como JAR [PPJF2017], WAR [PAWA2017], o la distribución sobre la base de metadatos sobre el proyecto, sin la necesidad de hacer ninguna secuencia de comandos en la mayoría de los casos.
- Gestión de Dependencias: Maven fomenta el uso de un repositorio central de archivos JAR y otras dependencias. Maven viene con un mecanismo que los clientes de su proyecto pueden utilizar para descargar cualquier JAR necesario para la construcción de su proyecto de un repositorio central de JARs. Esto permite a los usuarios de Maven reutilizar JAR a través de proyectos y fomenta la comunicación entre los proyectos para asegurar la compatibilidad con versiones anteriores.

7.3. RESTful

Los servicios web RESTful están diseñados para funcionar mejor en la Web. *Representational State Transfer* (REST) es un estilo arquitectónico que especifica restricciones, como una interfaz uniforme, que si se aplican a un servicio web inducen propiedades deseables, como: rendimiento, escalabilidad y modificabilidad, que permiten que los servicios funcionen mejor en la Web. En el estilo arquitectónico REST, los datos y la funcionalidad se consideran recursos y se accede mediante URI (*Uniform Resource Identifiers*), normalmente enlaces en la Web. Los recursos son representados utilizando un conjunto de operaciones simples y bien definidas. El estilo arquitectónico REST limita una arquitectura a una arquitectura cliente/servidor y está diseñado para utilizar un protocolo de comunicación sin estado (*Stateless*), típicamente HTTP [HTTP2017]. En el estilo de arquitectura REST, los clientes y servidores intercambian representaciones de recursos mediante el uso de una interfaz y un protocolo normalizados [ORAC2013].

Los siguientes principios fortalecen las aplicaciones RESTful a ser simples, livianas y rápidas:

- **Identificación de recursos mediante URI:** Un servicio web RESTful expone un conjunto de recursos que identifican los objetivos de la interacción con sus clientes. Los recursos se identifican mediante URI, que proporcionan un espacio de direccionamiento global para el descubrimiento de recursos y servicios [ORAC2013].

- **Interfaz uniforme:** Los recursos se manipulan utilizando un conjunto fijo de cuatro operaciones de creación, lectura, actualización y eliminación: PUT, GET, POST y DELETE. PUT crea un nuevo recurso, que puede eliminarse mediante DELETE. GET recupera el estado actual de un recurso en alguna representación. **POST** transfiere un nuevo estado a un recurso [ORAC2013].

- **Mensajes auto descriptivos:** los recursos se desacoplan de su representación para que su contenido pueda ser accedido en una variedad de formatos, como HTML, XML, texto sin formato, PDF, JPEG, JSON y otros. Los metadatos sobre el recurso están disponibles y se utilizan, por ejemplo, para controlar el almacenamiento en caché, detectar errores de transmisión, negociar el formato de representación adecuado y realizar autenticación o control de acceso [ORAC2013].

- **Interacciones con estado a través de hipervínculos:** Toda interacción con un recurso son sin estado; Es decir, los mensajes de petición son autónomos. Las interacciones de estado se basan en el concepto de transferencia de estado explicita. Existen varias técnicas de intercambio de estado, como sobre escritura de URI, cookies y campos de formularios ocultos. Estado se puede incrustar en mensajes de respuesta para señalar a estados futuros válidos de la interacción [ORAC2013].

7.4. OAuth 2.0

OAuth 2.0 Authorization Framework permite a una aplicación de terceros obtener acceso limitado a un servicio HTTP, ya sea en nombre del propietario de un recurso, orquestando una interacción de aprobación entre el propietario del recurso y el servicio HTTP o permitiendo que la aplicación de terceros obtenga acceso en su nombre propio [IETF2012].

Estrictamente hablando, el protocolo OAuth 2.0 es en realidad un protocolo de autorización y no un protocolo de autenticación. Debido a esto, OAuth 2.0 solo no puede proporcionar identidad federada. Sin embargo, cuando se utiliza de cierta manera y junto con otros protocolos, OAuth 2.0 puede proporcionar autenticación federada, que es un componente clave de los sistemas de identidad federados [BIHI2015].

El protocolo OAuth 2.0 define los siguientes cuatro roles:

Propietario del recurso: Una entidad capaz de conceder acceso a un recurso protegido. Cuando el propietario del recurso es una persona, se le conoce como usuario final.

Servidor de recursos: El servidor que aloja los recursos protegidos, capaz de aceptar y responder a las solicitudes de recursos protegidos mediante tokens de acceso.

Cliente: Una aplicación que hace solicitudes de recursos protegidos en nombre del propietario del recurso y con su autorización. El término "cliente" no implica ninguna característica particular de implementación (por ejemplo, si la aplicación se ejecuta en un servidor, un escritorio u otro dispositivo).

Servidor de autorización: El servidor emite tokens de acceso al cliente después de autenticar satisfactoriamente al propietario del recurso y obtener la autorización.

El servidor de autorización puede ser el mismo servidor que el servidor de recursos o una entidad independiente. Un único servidor de autorización puede emitir tokens de acceso aceptados por múltiples servidores de recursos [IETF2012].

La interacción entre los cuatro roles establecidos por el Protocolo OAuth 2.0 se muestra en la Figura 17. A continuación se describe brevemente la interacción entre estos cuatro roles para el proceso de autorización sobre un recurso.

(A) El cliente solicita autorización del propietario del recurso. La solicitud de autorización se puede hacer directamente al propietario del recurso (como se muestra), o preferiblemente indirectamente a través del servidor de autorización como intermediario.

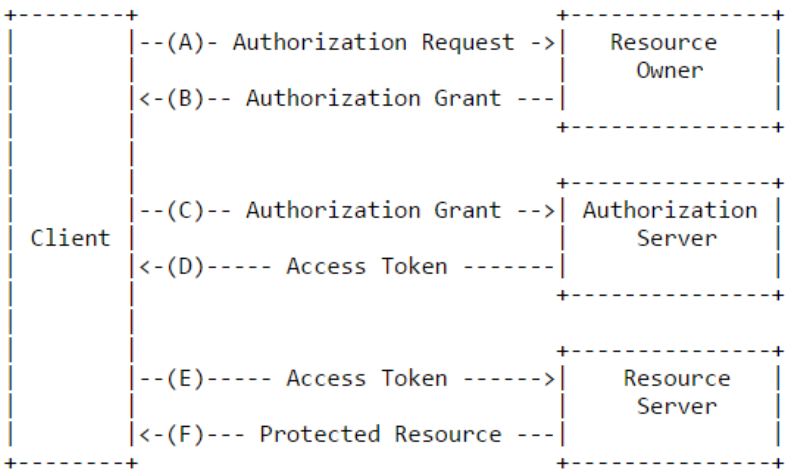

Figura 17. Diagrama de Interacción entre los Cuatro Roles del Protocolo OAuth 2.0

(B) El cliente recibe una concesión de autorización, que es una credencial que representa la autorización del propietario del recurso, expresada utilizando uno de los cuatro tipos de concesión definidos en esta especificación o utilizando un tipo de concesión de extensión. El tipo de concesión de autorización depende del método utilizado por el cliente para solicitar autorización y de los tipos admitidos por el Servidor de autorización.

(C) El cliente solicita un token de acceso mediante la autenticación con el servidor de autorización y la presentación de la concesión de autorización.

(D) El servidor de autorización autentica al cliente y valida la concesión de autorización y, si es válida, emite un token de acceso.

(E) El cliente solicita el recurso protegido desde el servidor de recursos y se autentica presentando el token de acceso.

(F) El servidor de recursos valida el token de acceso, y si es válido, ejecuta la solicitud.

IV-DESARROLLO DEL PROYECTO

En este capítulo se presenta las definiciones, artefactos y actividades relevantes realizadas en la ejecución del proyecto, así como las características principales de la Solución Informática.

1. Análisis y Comparativo de las Soluciones de Seguridad

En el capítulo 3 del Marco Teórico se relacionaron cuatro de las herramientas o aplicativos de software más representativos que ofrecen soluciones de seguridad. En el presente capítulo se realizará por medio de una tabla el comparativos de las características de estos productos con respecto a la solución propuesta.

Como se puede observar en la Tabla 1. se han seleccionado las características más representativas de las soluciones y herramientas encontradas que fueron descritas en el capítulo anterior y se ha realizado el comparativa de estas características con respecto a la solución **RSHIELD** que figura en la última columna, el cual es la propuesta del presente trabajo de grado.

Tabla Comparativa de Soluciones de Seguridad						
Características		**Herramientas o Soluciones**				
		OPENIAM Identity and Access Management	**WSO2** Identity Server	**FORGEROCK** Access Management	**ORACLE** Access Management	**RSHIELD** Solución de Seguridad en la Nube
Tipo Licencia	**OpenSource**	*x*	*x*			***x***
	Propietario			*x*	*x*	
Autorización		*x*	*x*	*x*	*x*	***x***
Autenticación		*x*	*x*	*x*	*x*	***x***
XACML		*x*	*x*	*x*	*x*	
SOA Security		*x*	*x*	*x*	*x*	
Social Identity (Integración Redes Sociales)			*x*		*x*	
Multi-tenant		*x*	*x*	*x*	*x*	***x***
SSO		*x*	*x*	*x*	*x*	***x***
Cloud		*x*	*x*	*x*	*x*	***x***
Roles Jerárquicos						**x**
Servicio de Notificaciones						**x**

Sucursales por Empresa					**x**
Restricción de Acceso por Franjas Horarias					**x**

Tabla 1. Comparativo de Soluciones de Seguridad.

Como resultado del proceso de documentación y análisis de soluciones de seguridad similares a la propuesta en el presente trabajo de grado se hallaron algunos productos OpenSource que cubren funcionalidades incluidas en el presente trabajo de grado. Con el fin de dar valor a la presente propuesta se hizo énfasis en características que no están cubiertas en las soluciones encontradas, o que solo están presentes en algunas de ellas.

La solución informática **RSHIELD** cubre al igual que las herramientas documentadas funcionalidades tales como: Servicios de Autenticación y Autorización, soporte para multi-tenencia, será una solución *cloud* desplegada en la nube, con soporte para SSO, entre otros. Con el fin de dar mayor flexibilidad a la solución propuesta y a los servicios ofrecidos por esta, se anexaron otras características a **RSHIELD** las cuales se pueden observar en la Tabla 1 en la última columna. Estas características se describen con más detalle a continuación.

- **Jerarquías de Roles:** Esta funcionalidad permitirá a las empresas cliente que hacen uso de la solución tener una mayor flexibilidad al momento de generar las estructuras de los roles de sus aplicaciones. Esto facilitará la configuración de nuevos roles al permitir heredar otros roles y sus respectivos permisos.

- **Servicio de Notificaciones:** Esta funcionalidad brindará la posibilidad a las aplicaciones cliente que hacen uso de la solución informática tener un sistema que permita enviar notificaciones a los usuarios finales de una aplicación.

- **Sucursales por Empresa:** Esta funcionalidad permitirá a las compañías clientes de la solución informática estructurar mejor la información de su compañía por medio de la configuración de sucursales por empresas. Esto brindará mayor flexibilidad a las compañías permitiendo realizar una segmentación de usuarios y roles de forma más granular tanto por empresa como por sucursal.

- **Restricción de Acceso por Franjas Horarias:** Esta funcionalidad permitirá a los representantes de las empresas cliente como usuarios de la solución informática, configurar permisos sobre las aplicaciones cliente basados en restricciones de tiempo o franjas horarias. Esto significa, que se podrán configurar restricciones a los usuarios finales de una aplicación cliente de forma que solo tengan acceso a una aplicación o a sus recursos en las franjas horarias o fechas establecidas en los permisos previamente establecidos.

2. Requerimientos y Funcionalidades

En el presente capítulo se relacionan los diferentes, actores, requerimientos y casos de uso identificados en la etapa de análisis de análisis de requerimientos y funcionalidades del proyecto con el fin de abarcar la problemática motivo del presente trabajo de grado. Finalmente se relacionan los casos de uso seleccionados para ser implementados en la solución desarrollada.

2.1. Actores

Para la solución informática se han definido cuatro roles principales: Administrador del Sistema (***SystemAdministrator***), Representante Empresa (***CompanyRepresentative***), Aplicación Cliente (***ApplicationClient***) y Usuario Aplicación (***ApplicationUser***). Los cuales se defines a continuación:

- **SystemAdministrator:** Es el usuario encargado del monitoreo, control y configuración de la solución informática. Es el encargado de realizar tareas de registro y administración de las cuentas de las diferentes empresas cliente que hacen uso de los servicios de seguridad ofrecidos por la solución informática.

- **CompanyRepresentative:** Es el usuario encargado de las tareas de registro de una empresa cliente en el sistema, el registro de sucursales y la respectiva configuración de las aplicaciones cliente de la empresa que harán uso de la solución informática. Otra de las tareas a realizar por el representante de una empresa incluye el registro de Roles por aplicación cliente, Roles Jerárquicos, Usuarios, Grupos de Usuarios y la respectiva asignación de permisos a los recursos de la aplicación.

- **ApplicationClient:** Hace referencia a una aplicación registrada en la solución informática por parte del Representante de una Empresa. Esta aplicación cliente hará uso de los servicios de seguridad para validar y verificar el correcto acceso a sus recursos por parte de un usuario final o **ApplicationUser**.

- **ApplicationUser:** Es el usuario final que hace uso o consume los recursos de una aplicación cliente.

2.2. Requerimientos Funcionales

A continuación, en la Tabla 2. se puede observar un listado en forma de resumen de los requerimientos recolectados para el desarrollo e implementación de la solución informática RSHIELD.

Listado Requerimientos RSHIELD		
Módulo funcional	**Código**	**Descripción del Requerimiento**

Administración de Empresas Cliente	*R01*	El sistema debe permitir el auto-registro y modificación de usuarios **representantes** de empresas
	R02	El sistema debe permitir luego del registro de un usuario representante de empresa la autorización por parte del administrador del sistema del acceso de este nuevo usuario al sistema
	R03	El sistema debe permitir la consulta, creación, modificación y eliminación de **empresas cliente**
	R04	El sistema debe permitir la consulta, creación, modificación y eliminación de **sucursales** por empresa cliente
Administración de Roles	*R05*	El sistema debe permitir la consulta, creación, modificación y eliminación de **roles** por aplicación cliente
	R06	El sistema debe permitir la consulta, creación, modificación y eliminación de roles jerárquicos
Administración de Aplicaciones Cliente	*R07*	El sistema debe permitir la consulta, creación, modificación y eliminación de **aplicaciones cliente.** Así mismo debe permitir el registro de las URL de páginas de la aplicación y las franjas horarias sobre la aplicación
	R08	El sistema debe permitir la asignación de URLs de páginas y franjas horarias a los roles
	R09	El sistema debe permitir la **asignación de roles a una aplicación cliente**
Administración de Usuarios	*R10*	El sistema debe permitir a los clientes la consulta, creación, modificación y eliminación de **usuarios de una aplicación cliente**
	R11	El sistema debe permitir la consulta, creación, modificación y eliminación de **grupos de usuarios de una aplicación cliente**
	R12	El sistema debe permitir a un usuario modificar la contraseña
	R13	El sistema debe permitir a un usuario restaurar la contraseña
	R14	El sistema debe permitir la asignación de roles a usuarios
	R15	El sistema debe permitir la asignación de roles a grupos de usuarios
Autenticación	*R16*	El sistema debe permitir a un usuario de una aplicación cliente autenticarse por medio de la solución informática
Autorización	*R17*	El sistema debe permitir validar si un usuario tiene los privilegios necesarios para acceder a un recurso determinado

Tabla 2. Tabla Resumen de Requerimientos

Para tener un poco más de detalle de los requerimientos de la solución informática estos se pueden encontrar en el Anexo 1.

2.3. Requerimientos No Funcionales

A continuación, en la tabla 3 se listan el conjunto de requerimientos no funcionales identificados en el proceso de levantamiento de requerimientos de la solución para cumplir con el objetivo general del presente trabajo de grado

Tabla de Requerimientos No-Funcionales		
Caracterización	**Código**	**Descripción**
Usabilidad	**RNF001**	La solución debe informar al usuario de una compañía cuando su registro sea autorizado
	RNF002	La interfaz web de administración de la solución debe ser accesible desde múltiples navegadores web.
	RNF003	La solución debe informar al usuario al momento que este haga modificaciones en el sistema
Modificabilidad	**RNF004**	La solución informática debe permitir realizar cambios en las funcionalidades del sistema sin impactar en múltiples partes del mismo.
Interoperabilidad	**RNF005**	La solución informática debe permitir la integración de la solución con aplicaciones de social media.
	RNF006	Los servicios de seguridad expuestos por la solución deben ser accesibles desde múltiples aplicaciones cliente implementadas en distintos lenguajes de programación.
	RNF007	La solución debe permitir la integración de sistemas externos para obtener los usuarios de cada compañía.
Seguridad	**RNF008**	La solución informática debe mantener cifrados los datos más sensibles, cómo la contraseña de los usuarios.

	RNF009	La solución debe asegurar que los datos no sean modificados de forma no autorizada. Como por ejemplo en la interacción entre una aplicación cliente y la solución.
	RNF010	La solución debe determinar la identidad de una persona, es decir validar que quien está intentando acceder a la solución es quien dice ser y que pertenece a la aplicación a la cual está intentando acceder.
	RNF011	Los datos transmitidos en la comunicación entre las aplicaciones cliente y los servicios de la solución informática deben estar cifrados con el fin de evitar la exposición de los clientes de la solución.
Escalabilidad	**RNF012**	La solución debe permitir la escalabilidad horizontal tanto de los servicios de seguridad como de la interfaz de administración web.
Performance	**RNF013**	La solución debe implementar servicios de caché con el fin de mejorar el rendimiento de los servicios de seguridad expuestos por la solución.
Multi-tenencia	**RNF014**	La solución debe soportar multi-tenencia en la misma base de datos
	RNF015	La solución debe permitir multi-tenencia a nivel de aplicación con el fin de ofrecer instancias diferentes y personalizables a cada tenant.
Extensibilidad	**RNF016**	La solución informática debe permitir el registro y validación de la autorización a URL de servicios REST de las aplicaciones cliente.

Tabla 3. Tabla de Requerimientos No-Funcionales

Debido a restricciones de tiempo se redujo el alcance del proyecto, por lo tanto algunos de los requerimientos no funcionales identificados y listados en la tabla 3, no fueron implementados en su totalidad o fueron implementados parcialmente. Algunos de los requerimientos no funcionales que no fueron implementados fueron: el RNF016, RNF015, RNF013, RNF007, RNF005.

Los requerimientos no funcionales no implementados y de mayor relevancia fueron relacionados y recomendados como posibles trabajos futuros y extensiones al presente trabajo de grado.

2.4. Casos de Uso

Para la selección de funcionalidades a implementar en la solución informática, se realizó un proceso de levantamiento de requerimientos los cuales fueron posteriormente seleccionados, agrupados y documentados como casos de uso. El listado completo de requerimientos se puede observar en el Anexo 1. Y el conjunto de Casos de Uso con mayor detalle se encuentran en el Anexo 2.

Listado de Casos de Uso				
Módulo	**Código**	**Nombre**	**Descripción**	**Requerimientos Soportados**
Administración de Empresas Cliente	**CU-01**	**Auto-Registro Representantes de Compañías**	La solución informática permite a un representante de una compañía (CompanyRepresentative) registrarse en la solución informática para posteriormente acceder a los servicios de seguridad.	R01
	CU-02	**Autorización de Registro de Representantes de Compañías**	La solución informática permite al administrador del sistema validar y posteriormente autorizar el registro de representante de una compañía en la solución informática	R02
	CU-03	**Administración de Empresas Cliente**	La solución informática permite la consulta, creación, modificación y eliminación de empresas cliente en la solución informática a los usuarios del sistema con rol CompanyRepresentative. Esto con el fin que las empresas puedan registrar la información correspondiente a la compañía y sus respectivas sucursales en caso que estas existan.	R03, R04
Administración de Aplicaciones Cliente	**CU-04**	**Administración de Aplicaciones Cliente**	La solución informática permite a un usuario con el rol **CompanyRepresentative** crear, editar, modificar y/o eliminar aplicaciones a una empresa cliente. Estas aplicaciones registradas en la solución informática y asociadas a una compañía o sucursal podrán hacer uso de los servicios de seguridad.	R07, R08
Administración de Roles	**CU-05**	**Administración de Roles**	La solución informática permite administrar los roles de las diferentes empresas clientes que hagan uso de la solución informática. Por lo tanto, se debe permitir a un usuario con el rol **CompanyRepresentative** crear, consultar, editar y/o eliminar roles y jerarquías de roles.	R05, R06
Administración de Usuarios	**CU-06**	**Administración de Usuarios**	Este caso de uso contiene las operaciones de creación, consulta, modificación y/o eliminación de usuarios y grupos de usuarios para una empresa cliente registrada en la solución informática. Con este caso de uso una empresa cliente podrá registrar los	R10, R11

			usuarios finales de las aplicaciones que tiene cada empresa.	
	CU-07	**Administración de Contraseñas**	Este caso de uso permite a los usuarios de la solución informática y a los usuarios finales de las aplicaciones de las diferentes empresas administrar sus contraseñas y realizar acciones como cambio y restauración de contraseñas.	R12, R13
Administración de Aplicaciones Cliente	**CU-08**	**Asignación de Roles**	Este caso de uso permitirá al usuario CompanyRepresentative realizar la asignación de los roles a los diferentes usuarios y grupos de usuarios de cada empresa cliente. Los permisos sobre las aplicaciones y los recursos de estas están basados en roles por tanto un usuario deberá tener roles o estar asociado a un grupo con roles para poder tener acceso a recurso determinado de una aplicación.	R09
Autenticación	**CU-09**	**Autenticación de Usuarios**	La solución informática permite a un usuario final autenticarse a través de la solución informática para poder acceder a una aplicación cliente registrada previamente en el sistema. Un usuario al acceder a una aplicación cliente es redireccionado a la solución informática junto con la información básica de la aplicación a la cual desea acceder. Allí, se le solicitan las credenciales para validar que sea un usuario registrado, que la aplicación a la cual desea acceder este registrada en la solución y por último que el usuario tenga permisos sobre la aplicación a la cual desea acceder	R16
Autorización	**CU-10**	**Autorización de Acceso a Recursos**	Este caso de uso se encarga de la validación, control y verificación del correcto acceso por parte de un usuario a un recurso o URL de una aplicación cliente. Un usuario luego de ser autenticado correctamente será direccionado a la aplicación requerida e intentará navegar y acceder a los distintos recursos con los que cuenta la aplicación. Es ahí donde se ejecuta este caso de uso, ya que el sistema debe verificar la petición de acceso al recurso por parte de un usuario y validar contra la solución informática si el usuario tiene o no los privilegios suficientes para acceder a dicho recurso	R17

Tabla 4. Listado de Casos de Uso **RSHIELD**

A continuación, en la Figura 18. se puede observar el diagrama de los casos de uso a implementar en la solución informática y su relación con los actores identificados.

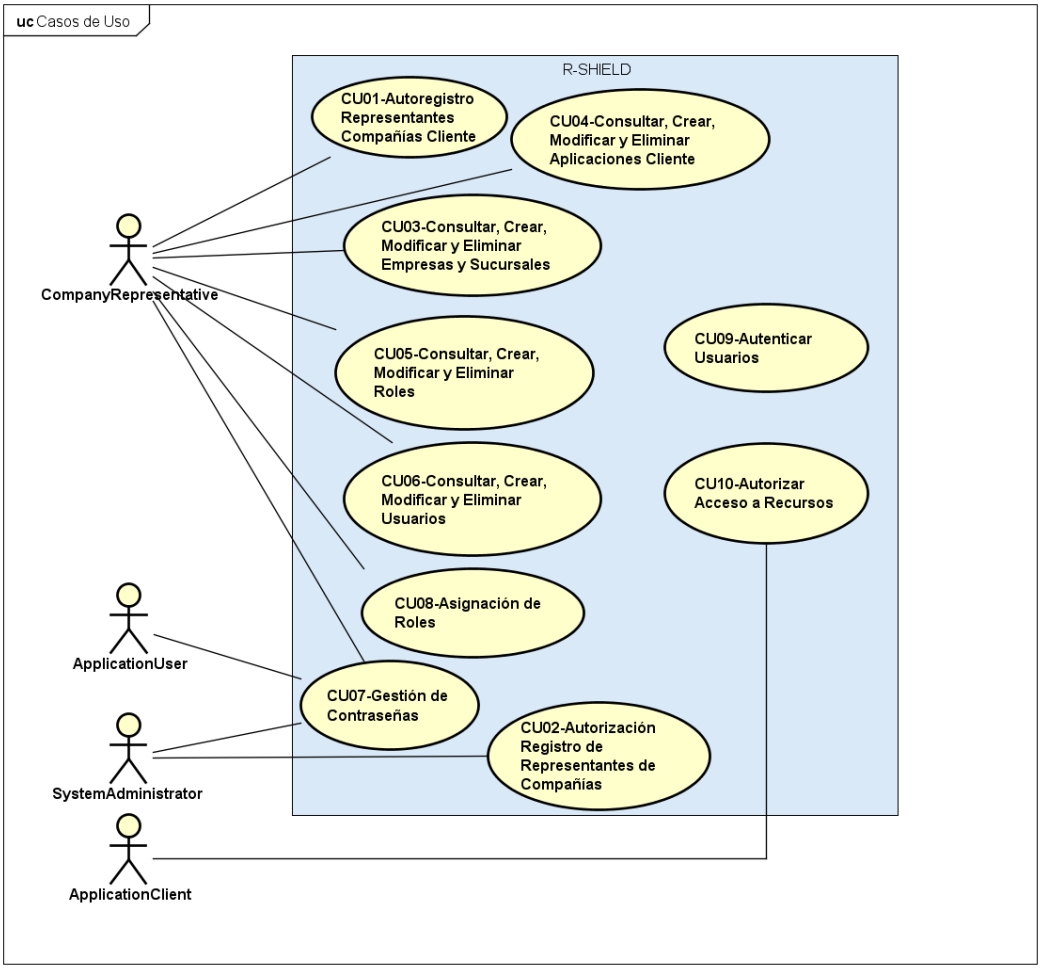

Figura 18. Diagrama de Casos de Uso de la Solución Informática

3. Tecnologías de la Solución

El desarrollo de la solución informática RSHIELD se apoyó en un conjunto de especificaciones, *APIs* y *Frameworks* los cuales brindan un conjunto de herramientas y buenas prácticas que apoyan el desarrollo de aplicaciones y para este caso que fueron fundamentales para conseguir el objetivo principal del presente trabajo de grado.

Lenguaje de Programación: Como lenguaje de programación principal de la aplicación se escogió Java en su versión 7.

Frameworks de Desarrollo: Como *Framework* principal de desarrollo de escogió *Java Enterprise Edition* versión 7 el cual viene con un conjunto de herramientas y utilidades que facilitan y mejoran el desarrollo de aplicaciones basadas en el lenguaje Java. Como Framework para el manejo del ciclo de vida del proyecto y manejo de dependencias se hará uso de Maven.

Por otra parte, para el desarrollo de los Servicios Web de seguridad que ofrecerá la solución informática a las aplicaciones cliente se utilizará RESTful y el Framework Apache Oltu[5].

Uno de los objetivos del presente trabajo de grado es que los servicios ofrecidos por la solución informática RSHIELD estén disponibles para diferentes aplicaciones basadas en diferentes lenguajes de programación. REST al estar basado en el protocolo HTTP permite crear servicios interoperables con múltiples lenguajes de programación. Por otra parte, Apache Oltu es una implementación en Java del protocolo OAuth 2.0. OAuth [IETF2012] permitirá implementar los servicios para el proceso de autenticación y autorización sobre los recursos URL de una aplicación por parte de los usuarios finales de las aplicaciones cliente de una empresa que haga uso de la solución informática RSHIELD. Para más detalle del protocolo OAuth referirse al capítulo 3-Marco Teórico a la sección 7.4.

Motor de Base de datos: para la implementación del presente trabajo de grado se escogió como motor de base datos PostgreSQL[6].

Para el desarrollo del presente trabajo de grado se optó por el uso de herramientas de software libre, ya que se pretende que la solución producto del presente proyecto quede disponible a la comunidad para su uso y modificación sin restricciones. Por otra parte, se optó por el lenguaje de programación Java y herramientas basadas en Java principalmente por motivos de conocimiento y manejo del lenguaje por parte de autor del presente trabajo de grado. Así mismo, otro motivo para la selección del lenguaje y de las herramientas en Java es la comunidad amplia de desarrolladores que ofrecen al público un conjunto de Frameworks que fueron fundamentales para el desarrollo del presente trabajo de grado.

4. Diseño y Arquitectura de la Solución

En este capítulo se documenta la arquitectura propuesta para la solución informática, las diferentes tecnologías, frameworks y los diferentes modelos de diseño utilizados en la implementación de la solución **RSHIELD**.

4.1. Modelo de Entidades de la Solución

A continuación, en la Figura 19. se puede observar el diagrama de entidades persistentes de la solución informática con los principales elementos identificados con el fin de cubrir todos los casos de uso y el objetivo principal del presente proyecto de grado que pretende desarrollar una

[5] https://oltu.apache.org/

[6] https://www.postgresql.org/

herramienta que ofrezca servicios de seguridad para las aplicaciones de las compañías que lo requieran.

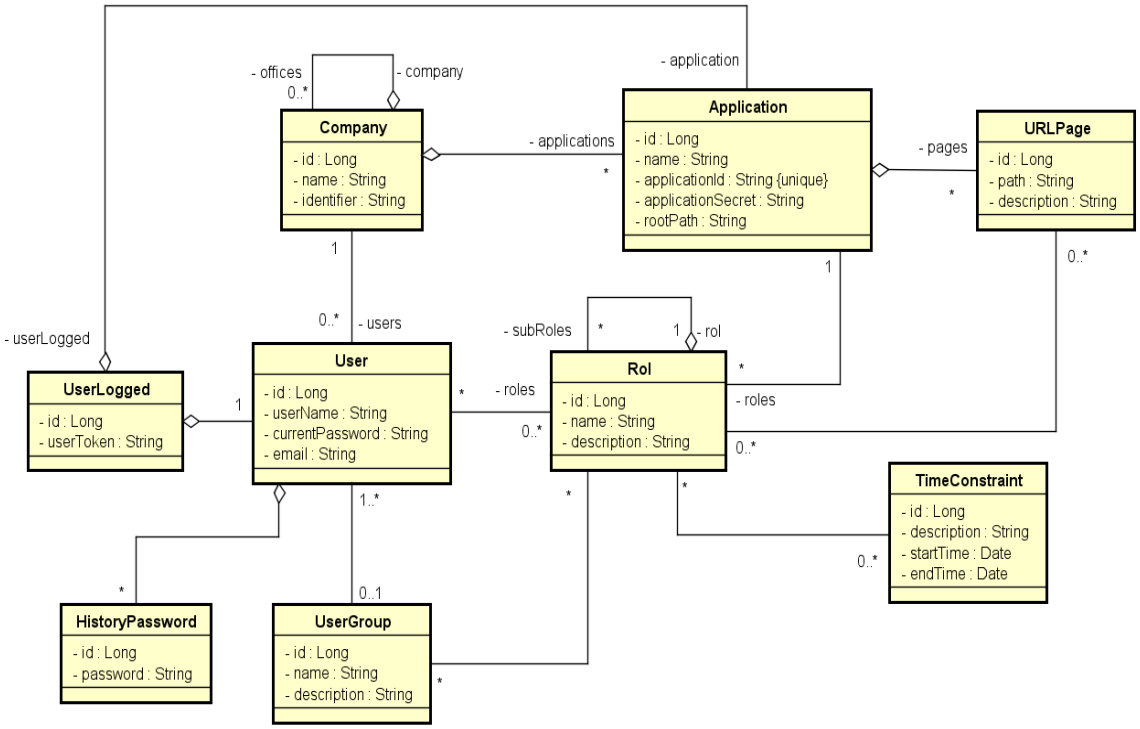

Figura 19. Modelo de Entidades Persistentes para R-SHIELD

En la Figura 19, se poder observar el modelo de entidades persistentes propuesto para la solución informática RSHIELD. En este modelo se presentan las entidades más significativas que harán parte del modelo de la solución informática como lo son: las Compañías, Usuarios, Grupos de Usuarios, Roles, Restricciones de Horario, Aplicaciones y Recursos o URL de aplicaciones. En este modelo se evidencian también algunos de los servicios que tendrá la solución como son las jerarquías de Roles, lo cual permitirá adquirir permisos de otros roles a través de la herencia de roles. También, se puede ver las jerarquías que podrán tener las compañías con sus sucursales lo que les permitirá reestructurar de mejor forma su información y sus usuarios permitiendo organizarlos por compañía o sucursal si se requiere.

4.2. Arquitectura

La solución propuesta en el presente trabajo de grado está compuesta o dividida en dos módulos principales: Uno con los servicios de cara al representante o compañía que desea hacer uso de los servicios de seguridad. La otra parte está compuesta por los servicios ofrecidos a las diferentes aplicaciones de las compañías cliente que hacen uso de los servicios de seguridad para proteger los recursos de las aplicaciones. Los dos partes o módulos se detallan a continuación.

La primera parte de la solución está compuesta de una interfaz gráfica de usuario para las tareas de administración a realizar por los representantes de las compañías cliente que hagan uso de la solución informática como se puede observar en la Figura 20. A través de esta interfaz los representantes de las compañías cliente podrán realizar tareas como:

- Registro de las compañías y sus respectivas sucursales.
- Configurar las aplicaciones de las empresas cliente que harán uso de la solución informática.
- Configuración de Usuarios y Grupos de Usuarios.
- Creación y configuración de Roles.
- Administración de los recursos a proteger: La solución informática permitirá protegerá las páginas (URL) de las aplicaciones de forma que estas solo puedan ser accedidas por usuarios con los privilegios adecuados y dentro de los rangos de horarios permitidos, ya que la solución permitirá establecer restricciones sobre las URL de las páginas basados en horarios de acceso.

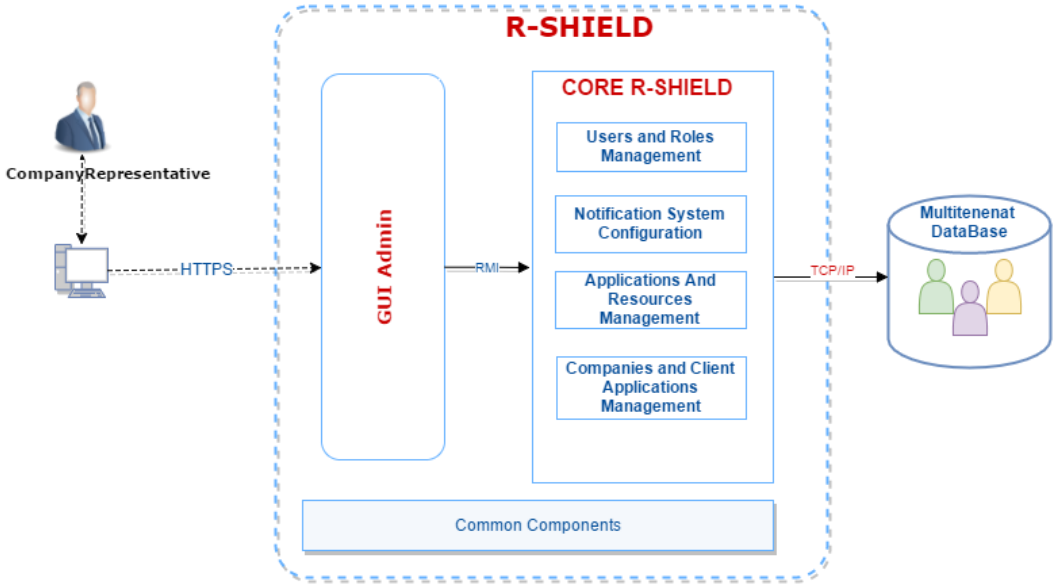

Figura 20. Arquitectura de Servicios para Representantes de Compañías

La segunda parte es la compuesta por el API de servicios web REST ofrecidos a las aplicaciones cliente registradas en la solución informática. Estas aplicaciones cliente podrán hacer uso de los servicios de seguridad para proteger sus recursos (URL). La protección de los recursos puede ser basada en restricciones sobre URL de las respectivas páginas o establecer restricciones basadas en franjas horarias.

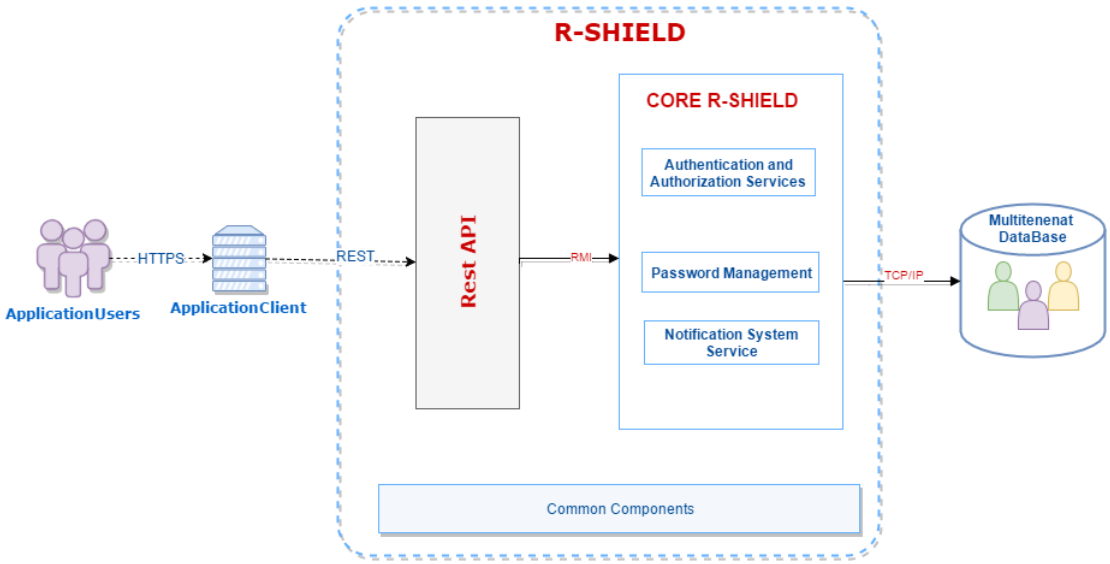

Figura 21. Arquitectura de Servicios para Aplicaciones Cliente

Como se puede observar en la Figura 21. se muestran los servicios ofrecidos por la solución informáticas a las aplicaciones cliente. Dentro de los servicios ofrecidos a las aplicaciones cliente se encuentran:

- **Autenticación**: La solución permitirá a las aplicaciones cliente validar al momento del ingreso de un usuario a la aplicación y no estar autenticados ser redireccionado por la aplicación cliente a la solución informática donde se les brindará el servicio de autenticación con el fin de que estos usuarios puedan acceder a la aplicación requerida de forma segura.

- **Autorización**: Luego de autenticado un usuario final de una aplicación cliente, la solución informática validará si el usuario autenticado tiene los privilegios para acceder a la aplicación cliente y sus recursos.

- **Notificaciones**: Un usuario representante de una compañía (CompanyRepresentative) tendrá acceso a un sistema de notificaciones a través de cual podrá crear y notificar a los usuarios finales de una aplicación. De esta forma, a través de una aplicación cliente que use los servicios de RSHIELD se podrán enviar mensajes y notificar a los usuarios finales conectados a la aplicación cliente.

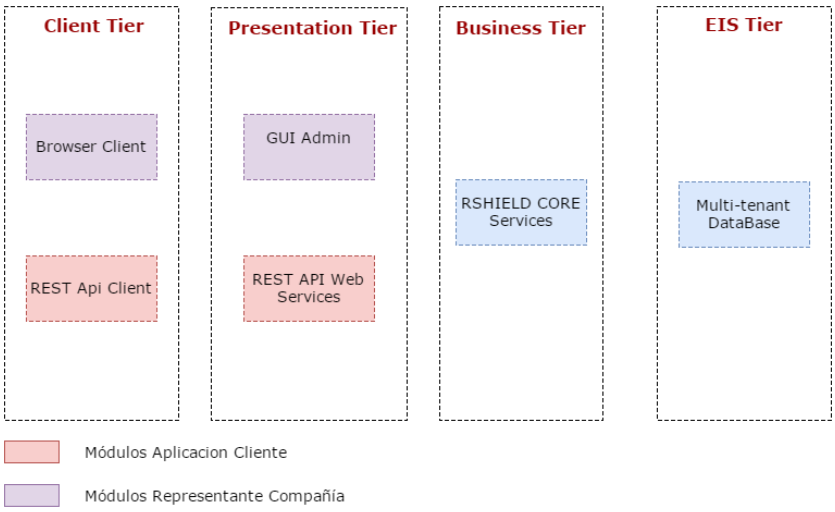

Figura 22. Arquitectura Multi-nivel para RSHIELD

El desarrollo de la solución informática R-SHIELD está basado en la plataforma de desarrollo JEE7 [GUPT2013] con una arquitectura multi-nivel (*multi-tier*) compuesta de cuatro niveles o *tiers.*

Como se puede observar en la Figura 21. La arquitectura propuesta para la solución informática está compuesta de los siguientes cuatro niveles o *tiers*. Un nivel Cliente (*Client Tier*), un nivel web (*Web Tier*), un nivel de lógica de negocio (*Business Tier*) y un nivel de Datos (*Enterprise Information System Tier*). A continuación, se describen cada uno de los niveles de la arquitectura propuesta, sus componentes y la razón de ser de cada uno.

- **Client Tier:** Por una parte, el nivel cliente está compuesta por un *browser* o navegador web por medio del cual el representante de la compañía interactúa con la solución. Por otra parte, están las aplicaciones clientes de los servicios REST expuestos por la solución informática.

- **Web Tier:** El nivel web al igual que el nivel de cliente tiene dos funcionalidades. Por un lado, están los componentes de interfaz de usuario para el registro y administración de las diferentes compañías clientes, sus sucursales, sus aplicaciones, roles, usuarios y demás tareas. Por otra parte, están los servicios REST expuestos por la solución informática que prestarán los servicios de seguridad ofrecidos a las distintas aplicaciones cliente de las compañías registradas en la solución informática R-SHIELD.

- **Business Tier:** El nivel de lógica de negocio contiene todos los componentes necesarios para la prestación de los servicios de la solución informática. En este nivel están todos los servicios de persistencia y lógica propia de la solución informática. Todos estos servicios están apoyados en los servicios provistos por el servidor de aplicaciones.

- **EIS Tier:** Este nivel hace referencia al nivel de almacenamiento de datos. El cual será una base de datos relacional que tendrá soporte multi-tenencia mediante el manejo de datos

desde la solución RSHIELD con el fin de almacenar la información y atender a múltiples compañías cliente que hagan uso de la solución informática.

4.3. Despliegue de la Solución

El despliegue inicial propuesto para la solución R-SHIELD como se puede observar en la Figura 23. Estará agrupado en tres *Tiers* o niveles los cuales son: el Tier Cliente, el Tier de Presentación y Lógica de negocio en uno y el Tier de EIS donde se encontrará el sistema de Base de Datos relacional de la solución.

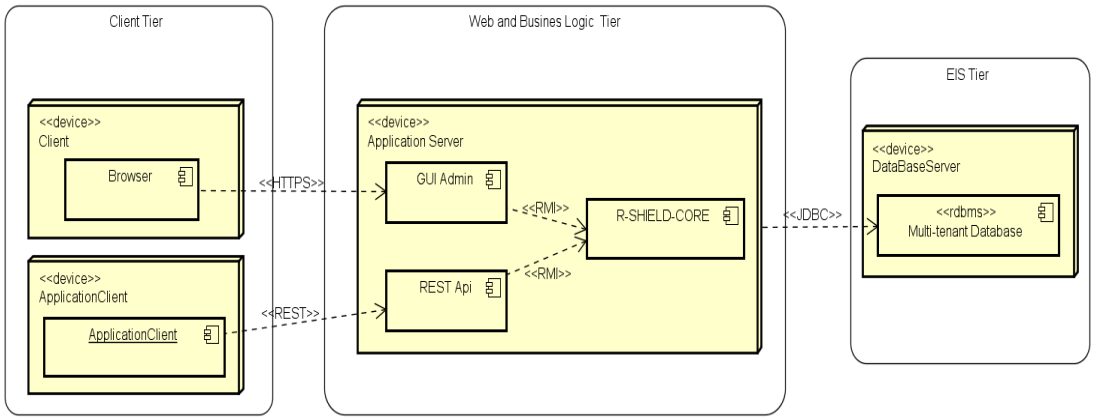

Figura 23. Diagrama de Despliegue de la solución R-SHIELD

4.4. Manejo aspecto Cloud y Multi-tenencia

Como se expresó en capítulos anteriores, el modelo multi-tenencia definido para el almacenamiento de datos de la solución informática RSHIELD, será un modelo Totalmente Compartido (Totally Shared) donde todos los tenants o clientes comparten la misma base de datos, tablas y esquemas. Este modelo de multi-tenencia permitirá aprovechar de forma más eficiente los recursos de la máquina donde se encuentre desplegada la solución informática y facilitará la administración de la misma.

Algunos desafíos que plantea la decisión tomada de usar un modelo Totalmente Compartido de multi-tenencia es la generación de backup de clientes específicos y la administración de información y en algunos casos la exposición de información de un tenant hacia otro tenant. En cuanto al manejo de información por parte de la solución informática RSHIELD se aclara que la solución no almacenará ningún tipo de información sensible corporativa ya que la solución solo requerirá la información respectiva que permita administrar las aplicaciones cliente de las compañías y prestar los servicios de seguridad respectivos ofrecidos.

Todas las restricciones mencionadas anteriormente fueron consideradas en el diseño de la solución, pero debido a motivos de alcance y de tiempo de ejecución del presente trabajo de grado se optó finalmente por un modelo de multi-tenencia Totalmente Compartido.

Por otro lado, en cuanto a la interfaz de administración o consola web de administración de la solución. Será una única instancia la encargada de atender todos los clientes de la solución. Es decir, no se generarán nuevas instancias independientes de la solución por cada nuevo cliente que se registre. Sin embargo se aprovechan el manejo de múltiples procesos livianos provistos y administrados por el servidor de aplicaciones (WildFly), uno por cada sesión de usuario e igualmente el pool de instancias de componentes de negocio.

4.5. Interacción Entre Una Aplicaciones Cliente y RSHIELD

A continuación, en la figura 24, se muestra el flujo de interacción de una aplicación cliente con los servicios de seguridad ofrecidos por la solución informática RSHIELD.

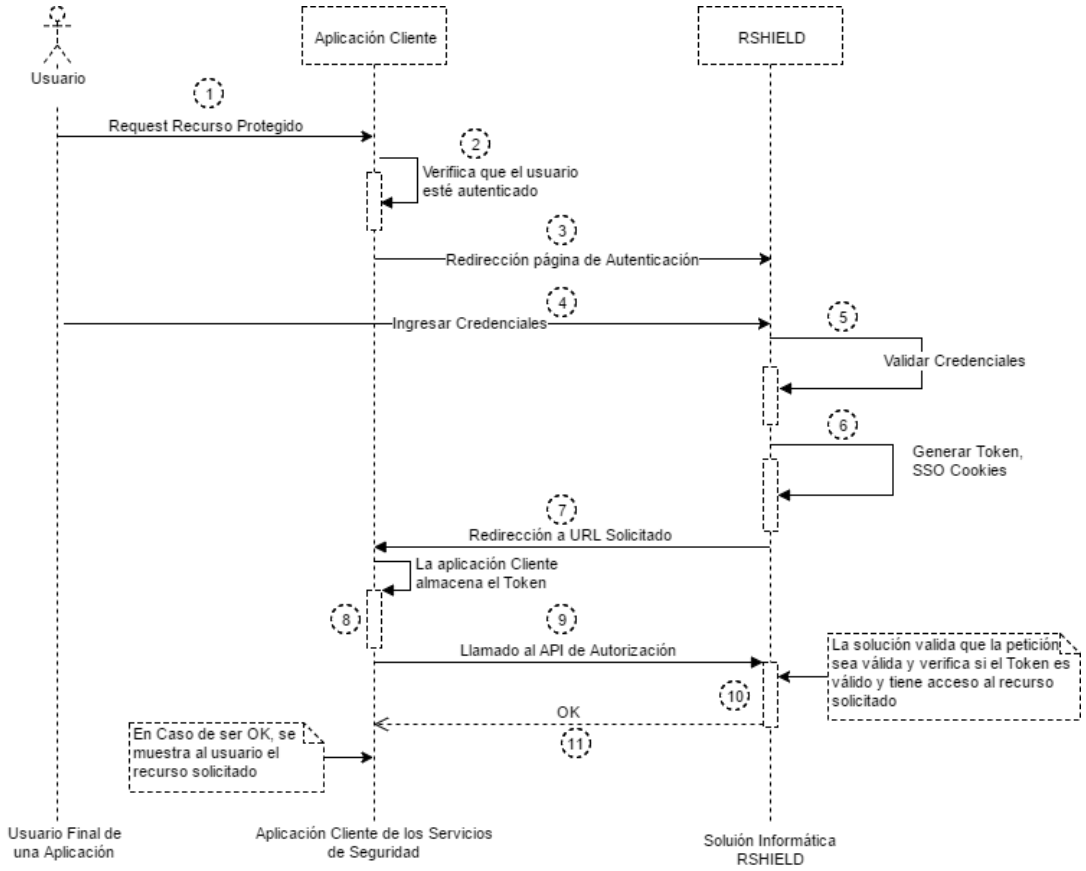

Figura 24. Flujo de interacción entre una Aplicación Cliente y RSHIELD

En la Figura 24 se muestra el flujo para acceder a un recurso (protegido) de una aplicación que está haciendo uso de los servicios de seguridad ofrecidos por la solución informática RSHIELD. Para el flujo de la imagen sólo se representa el flujo seguido para un usuario no autenticado previamente y adecuadamente registrado en la solución al igual que la aplicación cliente.

A continuación, se describen en mayor detalle cada uno de los pasos numerados en la anterior figura.

Paso 1: El primer paso del flujo representa el caso en el que un usuario final de una aplicación registrada previamente en la solución RSHIELD intenta acceder a un recurso protegido. En RSHIELD los recursos protegidos son páginas web de una aplicación cliente.

Paso 2: La aplicación cliente debe verificar mediante algún mecanismo que el usuario esté autenticado previamente ante la aplicación.

Paso 3: Partiendo del supuesto que el usuario no ha ingresado anteriormente a la aplicación, el sistema inmediatamente debe redireccionar al usuario a la pantalla de Login de la solución informática RSHIELD.

Paso 4: Luego de ser redireccionado a la pantalla de Login de la solución el usuario debe ingresar sus credenciales tales como nombre de usuario y password válidos. Hay otros datos requeridos en el proceso de autenticación de usuarios finales de aplicaciones, estos datos tales como el id de la aplicación a la cual se intenta loguear el usuario deben ser pasados como parámetro de la URL como se muestra en el capítulo 4.6.2.

Paso 5: Durante este paso la solución informática RSHIELD verifica que las credenciales ingresadas por el usuario pertenezcan a un usuario registrado en el sistema, que el usuario esté asociado a la aplicación cliente a la cual está intentando acceder.

Paso 6: Luego de verificadas las credenciales del usuario el sistema RSHIELD genera un token que será retornado a la aplicación cliente en el momento en que retorne control para que el usuario prosiga su acceso a una página web de la aplicación. Otro proceso importante que sucede durante este paso es la creación de la Cookie que servirá para identificar las peticiones de los clientes y activar el servicio de Single Sign On.

Paso 7: En este paso, luego de una autenticación exitosa el sistema redirecciona al usuario al recurso o URL de la aplicación cliente al cual estaba intentando acceder junto con el Token que fue generado y asociado al usuario. El token generado es retornado a la aplicación cliente mediante un parámetro de la URL de respuesta. Este token le permitirá posteriormente a la aplicación cliente identificarse ante la solución RSHIELD.

Paso 8: En este paso la aplicación cliente almacena mediante algún mecanismo el Token asignado por la solución al usuario que intenta acceder al recurso.

Paso 9: En este paso, luego de obtener el Token la aplicación cliente procede a invocar el servicio REST de la solución RSHIELD quien será finalmente quien le dé o no la autorización al usuario sobre el recurso. Durante este paso 9 la aplicación cliente debe enviar al Token a la solución.

Paso 10: Durante este paso la solución informática RSHIELD valida que la petición recibida sea válida. También valida que el token enviado tenga los permisos necesarios sobre el recurso solicitado, y responde un status OK o ERROR a la aplicación cliente.

Paso 11: Partiendo del supuesto que el usuario se autenticó correctamente ante la solución y cuenta con los permisos sobre el recurso solicitado, la aplicación cliente redirecciona finalmente al usuario al recurso solicitado.

4.6. Ilustración RSHIELD en Ejecución

En el presente capítulo se muestran instantes de la solución informática RSHIELD en ejecución. La solución fue probada haciendo uso de una aplicación prototipo construida con la finalidad de validar los respectivos servicios ofrecidos por RSHIELD.

Con el fin de realizar las pruebas de validación con el prototipo lo más reales posibles, la solución informática RSHIELD fue desplegada en un ambiente Cloud haciendo uso de los servicios ofrecidos por Amazon Web Services[7]. Esta instancia fue provista de forma gratuita por Amazon Web Services y tendrá una vida de alrededor de un año para ser utilizada. Esta instancia cuenta con 8GB de almacenamiento, 1GB de memoria RAM y un procesador para su operación.

4.6.1. Fase de Registro y Configuración

Para esta fase de registros de compañías cliente y su representante se parte del supuesto que la solución ya cuenta con un usuario Administrador del Sistema quién tendrá la responsabilidad de administrar los registros de compañías cliente y realizar la respectiva autorización con el fin de permitirle hacer uso de los servicios de seguridad ofrecidos por RSHIELD.

El primer paso a seguir para hacer uso de la solución informática es el registro de una compañía cliente junto con su usuario representante en la solución informática.

Al ingresar a la solución se puede observar un enlace “Registrarse para usar la Solución” que permitirá a los usuarios ir al formulario para registrarse en la solución.

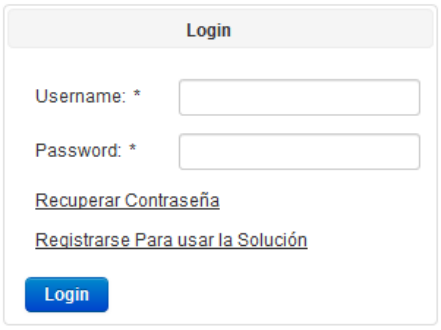

Figura 25. Pantalla de Login RSHIELD

[7] https://aws.amazon.com/ec2/

Luego de hacer clic en el enlace “Registrarse para usar la Solución”, el sistema redirecciona al usuario a la pantalla con el respectivo formulario de registro de una compañía y su usuario Representante de la Compañía.

Figura 26. Pantalla registro de Clientes

Luego de registrado el usuario Representante de un Compañía el sistema RSHIELD le notificará vía correo electrónico que su registro fue validado correctamente por el Administrador del Sistema y podrá ingresar a la solución para realizar del registro de todas las aplicaciones cliente que desea que hagan uso de los servicios de seguridad de RSHIELD.

Dentro de los anexos de encuentran los manuales respectivos con mayor detalle del proceso de instalación, configuración y usos de la solución y sus respectivos servicios de seguridad.

Una vez sea notificado el usuario (Representante de Compañía) podrá acceder a los servicios de la solución tal como se puede observar en la siguiente figura.

Figura 27. Pantalla bienvenida RSHIELD.

Como se muestra en la Figura anterior, una vez el nuevo usuario Representante de una compañía y usuario de la solución ingresa, se presenta al lado izquierdo el menú de opciones con las que cuenta el usuario. Dentro de este conjunto de opciones se encuentra el registro de Aplicaciones, registro de Usuarios, registro de Roles, Restricciones Horarias y demás servicios ofrecidos por RSHIELD.

Dentro del anexo 7, se encuentra a mayor detalle la descripción de casa una de las funcionalidades de la solución informática RSHIELD. También, en este anexo 7 se explica el uso de casa funcionalidad.

4.6.2. Autenticación de Usuarios Finales

En el momento en que un usuario ingresa a una aplicación cliente por primera vez, y solicita acceso a un recurso protegido de la aplicación que en este proyecto son URL de páginas web, la aplicación debe validar que exista un Token en la aplicación cliente, de no ser así, la aplicación cliente debe direccionar al usuario a la pantalla de autenticación de la solución informática como se observa en la siguiente imagen.

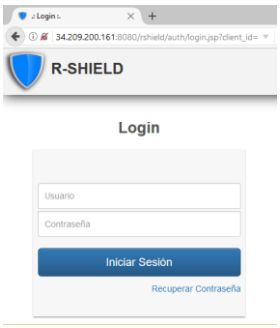

Figura 28. Pantalla Login para Usuarios Finales de Aplicaciones

Al momento de una aplicación redireccionar al usuario a la solución informática para realizar el proceso de autenticación, la aplicación cliente debe anexar en la URL algunos datos extra utilizados en el proceso de autenticación del usuario.

A continuación, se muestra un ejemplo de una URL de autenticación para una aplicación llamada ACME.

```
http://host:port/rshield/auth/login.jsp?client_id=fUGE8EZoj7n2jgvr&application_name=ACME&redirect_uri=http://localhost:8180/acme/admin/users.xhtml
```

- **client_id:** El client_id hace referencia al ID autogenerado para la aplicación en el momento de registro.
- **application_name:** El application_name es el nombre de la aplicación registrada y a la cual el usuario se está intentando autenticar.
- **redirect_uri:** El redirect_uri hace referencia a la URL de la aplicación cleinte a la cual será redireccionado el usuario luego se haber sido correctamente autenticado.

Una vez el usuario haya sido autenticado exitosamente la solución informática RSHIELD redirecciona al usuario a la URL provista en el parámetro **redirect_uri**. A continuación, se muestra un ejemplo del proceso de redireccionamiento a una aplicación cliente luego de autenticado satisfactoriamente un usuario.

```
http://host:port/acme/admin/users.xhtml?access_token=aW9uQ29kZSI-
sImV4cCI6OTAwMDAwLCJpYXQiOjE0OTUyNDMwMjEyNjF9.LM3KurIL
```

Como se ve en el texto anterior la solución redirecciona al usuario a la URL pasada en el parámetro **redirect_uri** y lo hace junto con el Token generado por la solución RSHIELD en el momento de la autenticación. Este token generado será enviado por la aplicación cliente para identificar posteriormente las peticiones que una aplicación cliente hace a la solución informática haciendo uso de los servicios REST para la autorización de acceso a recursos de la aplicación en la cual se encuentra el usuario navegando.

4.6.3. Validación de Autorización Sobre Recursos

Luego del proceso de configuración y el proceso de autenticación de un usuario por parte de la solución viene el proceso de autorización de usuarios con el fin de verificar si un usuario tiene los privilegios necesarios para acceder a un recurso de una aplicación cliente.

El proceso de autorización de usuarios sobre recursos se hace a través del API REST de autorización expuesto por RSHIELD. La finalidad de este conjunto de servicios expuestos por la solución en formato REST es permitir a múltiples tipos de aplicaciones cliente poder interactuar con la solución informática y poder hacer uso de los servicios de seguridad ofrecidos.

El principal servicio expuesto por RSHIELD es el servicio de autorización; este servicio permitirá a las aplicaciones cliente validar contra la solución informática si un usuario tiene permisos sobre un recurso o URL específico. La petición sobre este recurso va acompañada del Token generado en el proceso de autenticación.

```
POST /rshield/authz/services/authorize HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/plain
Authorization: Bearer eyJhbGciOiJSUzI1NiIsImFjY2Vzc190b2tlbiI6Im94aVhIR2NWME9SenNPblJ6QjJONlRWRUVVWkVpMXpzIn0.
Content-Length: 18
Host: remotehost:port

/admin/users.xhtml
```

Figura 29. Ejemplo Request Servicio de Autorización

En la figura anterior se muestra un ejemplo de una petición al servicio de autorización ofrecido por la solución para validar si un usuario tiene acceso al recurso enviado en el cuerpo de la petición, para este ejemplo el recurso al cual se estaría intentando acceder sería /admin/users.xhtml.

Además del servicio de autorización la solución cuenta con otros dos servicios REST. Uno que permite consultar las páginas públicas (las páginas públicas son páginas a las cuales todos los usuarios de una aplicación tienen acceso independiente de rol que el usuario tenga) de una

aplicación y la otra para consultar el listado de recursos y las franjas horarias en las cuales un usuario tiene acceso. Estos dos servicios mencionados podrían ser extendidos con el fin de implementar servicios locales de cache en la aplicación cliente con el fin de disminuir la carga al servicio de autorización. Además de los servicios desarrollados inicialmente la solución informática RSHIELD podría ser extendida con el fin de exponer a modo de servicio REST los otros servicios ofrecidos por la solución como lo es la gestión de Usuarios, Roles, Compañías y Sucursales entre otros.

5. Validación de la Solución Informática

El proceso de validación de la solución informática producto del presente trabajo de grado se realizó desde dos frentes. Uno fue a través del desarrollo de prototipos de prueba y la otra a través de la socialización de la solución con el fin de validar a través de otros puntos de vista el impacto y/o utilidad del presente trabajo de grado.

Para la parte de validación a través de prototipos se desarrollaron dos aplicaciones cliente demo. Estas aplicaciones fueron desarrolladas, la primera en Java con JSF 2.0 y la segunda en JavaScript haciendo uso de AngularJS. Estas dos aplicaciones permitieron probar el nivel de interoperabilidad de la solución con aplicaciones desarrolladas en diferentes tecnologías y lenguajes de programación. Los servicios de seguridad de la solución al ser desarrollados haciendo uso de servicios REST permitió la fácil integración con estos prototipos lo cual demostró una rápida y fácil adopción de la solución con respecto a aplicaciones desarrolladas en lenguajes y plataformas diversas. Los diferentes prototipos utilizados para esta etapa se encuentran en los anexos asociados al presente trabajo de grado.

La socialización de la solución informática se realizó con una empresa de tecnología llamada Assert Solutions S.A.S. Esta compañía fue fundada inicialmente en Argentina y actualmente cuenta con una sede en Colombia hace alrededor de 5 años. La socialización del presente trabajo de grado se realizó con dos personas de esta compañía, el arquitecto de soluciones y el director de la sede en Colombia. Como resultado de esta socialización se resaltó el potencial de la presente propuesta, y al mismo tiempo se detectaron algunos elementos que deben ser tenidos en cuenta para posibles trabajos futuros que permitirían a la solución desarrollada ser más efectiva.

Como resultado de la validación de la solución informática RSHIELD con personas ajenas al desarrollo del trabajo se pudieron evidenciar un conjunto temas a tener en cuenta los cuales serán mencionados en la sección de trabajos futuros y conclusiones.

En este proceso de socialización con terceros se realizó una encuesta la cual se encuentra en el anexo 3 del presente trabajo de grado.

V-CONCLUSIONES Y TRABAJOS FUTUROS

1. Conclusiones

Con las pruebas realizadas a través de los prototipos se pudo evidenciar que la solución producto del presente trabajo de grado permite que aplicaciones desarrolladas en lenguajes de programación diferentes se acoplen rápidamente a la solución RSHIELD y puedan hacer uso de sus servicios de seguridad para proteger los recursos de sus aplicaciones.

Los servicios REST que se han popularizado tanto en la actualidad y que fueron utilizados para implementar el presente trabajo de grado, son una herramienta eficaz para lograr la interoperabilidad debido a que la mayoría de los lenguajes de programación y plataformas tiene soporte para esta tecnología. Se debe tener en cuenta y dependiendo del nivel de criticidad de la información se deben emplear técnicas de cifrado que permitan proteger el transporte de la información de forma segura.

Las soluciones que permiten ofrecer servicios multi-tenencia y en la nube como el presente trabajo de grado se convierten en una opción atractiva para aquellas empresas que no cuentan con la infraestructura y conocimiento para implementar los servicios requeridos y les dan la posibilidad a estas empresas de acceder a estos servicios de manera más rápida y eficiente.

El presente trabajo de grado representa una opción para la implementación de servicios de seguridad con el fin de proteger los recursos o URL de las aplicaciones de las compañías. Al ser un producto de software libre el presente trabajo de grado representa una oportunidad para aquellos interesados en la mejora o extensión de la presente solución informática.

2. Trabajos Futuros

La solución informática resultado del presente trabajo de grado presenta una propuesta de cómo ofrecer servicios de seguridad para proteger recursos URL de aplicaciones web de compañías, pero a su vez este trabajo de grado puede ser extendido y mejorado con el fin de ofrecer nuevos y mejores servicios de seguridad. A continuación, se describen algunos posibles trabajos futuros que podrían potencializar los servicios ya prestados por la solución RSHIELD.

Ya que los usuarios finales de las aplicaciones cliente de la solución RSHIELD deben ser creados manualmente por la compañía o el representante de la compañía, un módulo que permita la integración de usuarios ya existentes en otros sistemas de información como los LDAP y Directorios Activos con la solución aumentaría considerablemente los tiempos de configuración y utilización de la solución por parte de una compañía cliente interesada en los servicios de seguridad ofrecidos por RSHIELD.

De la mano con el requerimiento no funcional RNF005 y con el fin de ampliar la integración de la solución con otros sistemas, se propone como trabajo futuro agregar un módulo de integración de usuarios desde los APIs expuestos por redes sociales.

Se podría extender la solución para que permita no sólo proteger recursos o URL de aplicaciones como está hasta el momento, sino que permita el registro y protección de URL de servicios web como pueden ser servicios web REST con el fin de implementar lo establecido en el Requerimiento no funcional RNF015.

La presente propuesta podría ser extendida no solo para prestar servicios de seguridad para la protección de recursos URL de aplicaciones web, sino que podría ser utilizado para proveer servicios de seguridad en otros sistemas como pueden ser los sistemas de control de acceso de las compañías.

Otro trabajo futuro que podría realizarse sobre el presente trabajo de grado es la extensión del componente de notificaciones para que permita el envío de notificaciones en tiempo real a los usuarios de las aplicaciones finales. Actualmente el sistema solo cuenta con notificaciones vía correo electrónico para notificar a los usuarios en la fase de registro y cambio o restauración de contraseña.

Basado en las recomendaciones realizadas en la etapa de evaluación y socialización de la solución con terceros de la mano con el conjunto de requerimientos no funcionales definidos, nace otro trabajo futuro que se podría anexar a la presente solución que consiste en evaluar el comportamiento de la solución RSHIELD ante el evento de alta carga de usuarios finales y realizar los ajustes necesarios que ayuden a la solución a mejorar su desempeño. Esto con el fin de dar solución al requerimiento no funcional RNF012 de escalabilidad y el RNF013 de servicios de caché en la solución.

VI-REFERENCIAS

[AGRA2011] Divyakant A., Wen-Syan L. K. Selcuk C., New Frontiers in Information and Software as Services: Service and Application Design Challenges in the Cloud, Springer-Verlag Berlin and Heidelberg GmbH & Co. K, pp.6-14, 2011.

[ALAI2013] Diego Martín Alaimo, Proyectos ágiles con Scrum : flexibilidad, aprendizaje, innovación y colaboración en contextos complejos. Primera Edición. - Ciudad Autónoma de Buenos Aires : Kleer, 2013.

[ANSI2003] American National Standards Institute, Inc., Role Based Access Control. BSR INCITS 359, 2003

[BECK2004] Kent Beck, Extreme Programming Explained, Embrace Change. Second Edition, Addison-Wesley, 2004.

[BECK2005] Kent Beck, Cynthia Andres. Extreme Programming Explained: Embrace Change, Second Edition. Pearson Education, Inc. 2005

[BECK2016] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas, Manifiesto por el Desarrollo Ágil de Software. [Online] http://agilemanifesto.org/iso/es/manifesto.html

[BIHI2015] Charles Bihis, Mastering OAuth 2.0. Packt Publishing, 2015

[DEYO2008] Jeremy Deyo, Software as a Service (SaaS), A look at the migration of applications to the web. Virginia Commonwealth University, pp.4-8, 2008.

[EAST2012] Chuck Easttom, Computer Security Fundamentals. Segunda Edición, Pearson 2012

[FERR1992] Ferraiolo, D., and D. R. Kuhn, Role-Based Access Control, Proceedings of the NIST-NSA National (USA) Computer Security Conference, 1992, pp. 554–563.

[FERR2007] David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli. Role-Based Access Control, Second Edition 2007 ARTECH HOUSE, INC. 685 Canton Street Norwood, MA 02062

[FIELD2000] Roy Thomas Fielding, Architectural Styles and the Design of Network-based Software Architectures, University Of California Irvine, 2000

[FORG2016] FORGEROCK, Access Management, [Online]. Available: https://www.forgerock.com/#what, 2016

[FOWL2003] Martin Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, Addison-Wesley Professional, p.79, 2003.

[FRAN2011] María Consuelo Franky, Victor Manuel Toro C, CincoSecurity: Automating the Security of Java EE Applications with Fine-Grained Roles and Security Profiles. International Journal On Advances in Security, 2011 no 3&4 (IARIA Journal), July 1, 2011 to December 31, 2011 – Online. ISSN: 1942-2636 http://www.thinkmind.org/index.php?view=article&articleid=sec_v4_n34_2011_10

[GOOG2016] Google, Google Identity Platform. [Online]. Available: https://developers.google.com/identity/

[GUPT2013] Arun Gupta, Java EE 7 Essentials. O'Reilly Media, Inc. 2013

[HURW2012] Judith Hurwitz, Marcia Kaufman, Dr. Fern Halper, Cloud Services for Dummies. Wiley, John Wiley & Sons, Inc., 2012

[IETF2012] Internet Engineering Task Force (IETF), The OAuth 2.0 Authorization Framework. [Online] https://tools.ietf.org/html/rfc6749. 2012

[KAVI2014] Michael Kavis, Architecting The Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, And IaaS). Wiley, p.59, 2014.

[MART2016] Antoni Martínez Ballesté, Agustí Solanas, Jordi Castellà Roca. Identificación, Autenticación y Control de Acceso, Universitat Oberta de Catalunya. [Online] https://www.exabyteinformatica.com/uoc/Dactiloscopia/Identidad_digital/Identidad_digital_(Modulo_1).pdf

[MELL2009]	Peter Mell and Tim Grance, The NIST Definition of Cloud Computing - Recommendations of the National Institute of Standards and Technology. NIST Publicación Especial 800-145. 2011
[MICR2017]	Mircrosoft, .NET, [Online]. Available: https://www.microsoft.com/net, 2017
[OASI2013]	OASIS, eXtensible Access Control Markup Language (XACML) Version 3.0. [Online] http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf 2013
[OPEN2016]	OpenIAM, OpenIAM Identity and Access Management. [Online]. Available: http://www.openiam.com/products/identity-manager/idm-overview/
[OPEN2016]	The Open Group, Single Sign-On. [Online] http://www.opengroup.org/security/sso/
[ORAC2013]	Oracle, The Java EE 6 Tutorial, [Online] http://docs.oracle.com/javaee/6/tutorial/doc/javaeetutorial6.pdf. 2013
[ORAC2014]	Oracle, Java Platform, Enterprise Edition, The Java EE Tutorial Release 7, 2014
[ORAC2014]	ORACLE, JAAS Authentication Tutorial, [Online]. Available: http://docs.oracle.com/javase/7/docs/technotes/guides/security/jaas/tutorials/GeneralAcnOnly.html, 2014
[ORAC2016]	ORACLE, Access Management, [Online]. Available: http://www.oracle.com/technetwork/middleware/id-mgmt/overview/accessmanagementdatasheet-2538966.pdf?ssSourceSiteId=ocomen, 2016
[PHIF2004]	Bill Phifer, DAR Basics: Applying Decision Analysis and Resolution in the Real World. Edition, 2004.
[PHP2017]	PHP, [Online]. Available: https://secure.php.net/, 2017
[RUMB2004]	James Rumbaugh, Ivar Jacobson, Grady Booch, The Unified Modeling Language Reference Manual. Second Edition, Addison-Wesley, p.77, 2004.

[SAND1998] Ravi Sandhu, Role-based Access Control. Advances in Computers, Vol. 46, 1998.

[SCRU2016] SCRUMstudy, A Guide to the SCRUM BODY OF KNOWLEDGE, SCRUMstudy 2016 Edition

[SOSI2011] Barrie Sosinsky, Cloud Computing Bible. 1st Edition, Wiley, 2011.

[W3C2004] W3C. Web Services Architecture. W3C Working Group Note 11 February 2004. [Online] https://www.w3.org/TR/ws-arch/#whatis

[WELL2009] Don Wells, Extreme Programming: Release Plan. [Online]. Available: http://www.extremeprogramming.org/rules/commit.html, 2009

[WSO22016] WSO2, Identity Server, [Online]. Available: http://wso2.com/products/identity-server, 2016

[HTTP2017] Hypertext Transfer Protocol -- HTTP/1.1, [Online]. Available: https://tools.ietf.org/pdf/rfc2616.pdf

[PPJF2017] Packaging Programs in JAR Files. [Online]. Available: https://docs.oracle.com/javase/tutorial/deployment/jar/

[PAWA2017] Packaging Web Archives, [Online]. Available: https://docs.oracle.com/javaee/7/tutorial/packaging003.htm

[MAVE2017] Apache Maven, [Online]. Available: https://maven.apache.org

VII-ANEXOS

Anexo 1. Requerimientos y Casos de Uso

En este anexo están relacionados todos los requerimientos identificados para la solución informática. Estos requerimientos están documentados y agrupados en casos de uso. Este anexo puede ser descargado desde el sitio web del trabajo de grado.

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD/anexos.html

Anexo 2. Documento de Casos de Uso

En este anexo está el documento final de los casos de uso identificados e implementados en la solución informática. Los cuales pueden ser descargados desde el sitio web del trabajo de grado.

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD/anexos.html

Anexo 3. Validación

En este anexo se encuentran los documentos de las encuestas que se realizaron a terceros en la etapa de validación y socialización de la solución. Este anexo puede ser descargado desde el sitio web del trabajo de grado.

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD/anexos.html

Anexo 4. Código Fuente de Solución Informática

En este anexo se encuentra el código fuente de la solución implementada. Al igual que el código fuente respectivo de los prototipos desarrollados para probar la solución. El código fuente puede ser descargado del sitio web del trabajo de grado.

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD/anexos.html

Anexo 5. Código Fuente Prototipo Java

En este anexo se encuentra el código fuente de la aplicación web prototipo y el filtro desarrollados en Java para validar la solución informática. El código fuente del presente anexo poder ser descargado del sitio del trabajo de grado.

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD/anexos.html

Anexo 6. Código Fuente Prototipo JavaScript

En este anexo se encuentra el código fuente del prototipo de la aplicación desarrollada en AngularJS que sirvió como elemento de validación de la solución informática. El código fuente respectivo de este anexo puede ser descargado del sitio del trabajo de grado.

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD/anexos.html

Anexo 7. Manual de Usuario

En este anexo se encuentra el manual de usuario respectivo con las instrucciones para hacer uso de la solución informática producto del presente trabajo de grado. Este manual puede ser descargado desde el sitio web del trabajo de grado.

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD/anexos.html

Anexo 8. Manual Técnico y de Instalación

Este anexo contiene la documentación respectiva con la orientación para realizar el despliegue de la solución informática. También, se encuentra el manual técnico desde se documenta los pasos a realizar para hacer uso de los servicios de seguridad de la solución una vez desplegada y configurada. Este documento puede ser descargado desde la página del trabajo de grado.

http://pegasus.javeriana.edu.co/~PA1710-8-RSHIELD/anexos.html