Módulo de autenticación SAML2 para DokuWiki

Máster Oficial de Software Libre. Web y comercio electrónico

Autor: Jorge Hervás Vigueras
Consultor: Cándido Rodríguez Montes

09/06/11

Resumen del proyecto

Este Proyecto Final del Máster Oficial de Software Libre, *Módulo de autenticación SAML2 para DokuWiki,* consiste en el desarrollo de un módulo que permite integrar la aplicación web DokuWiki con una infraestructura de autenticación y autorización basada en SAML2 (un sistema Single Sign-On). Concretamente la infraestructura SAML2 a la que se hace referencia consiste en la instalación de la aplicación SimpleSAMLphp con el consiguiente despliegue de un Proveedor de Servicios y un Proveedor de Identidad.

La integración del sistema wiki con la infraestructura SAML2 se ha debido hacer mediante la API proporcionada por la aplicación SimpleSAMLphp, lo que permite que un usuario de DokuWiki pueda introducir sus credenciales y que el Proveedor de Identidad correspondiente valide y responda con la información de sesión del usuario. Esta información de sesión le es suministrada en primer lugar al Proveedor de Servicios que a su tiempo se la entrega de vuelta al módulo DokuWiki aquí implementado.

Para llevar a cabo todo esto se ha debido analizar en que parte de la aplicación DokuWiki se debía añadir el código para proporcionar la nueva funcionalidad, y por otra parte se ha debido comprender el mecanismo del proceso de autenticación de SimpleSAMLphp/SAML2 para saber de que forma se debía implementar la solución de integración de los dos sistemas.

Índice de contenidos

1. Introducción

El presente Proyecto Final del *Máster Oficial de Software Libre* de la UOC corresponde al área Web y e-commerce. Las tecnologías claves que definen al proyecto son: php, dokuwiki y simplesamlphp. Todas ellas son proyectos de código abierto.

Antes de adentrarnos en el estudio de viabilidad se verá un breve estado del arte y se presentaran los principales proyectos de terceros relacionados con nuestro proyecto para situarnos en contexto. Será en el estudio de viabilidad, sin embargo, donde se presente un estado del arte más extenso (en "Estudio de la Situación Actual") y los objetivos del proyecto de carácter más técnico.

En lo referente a la estructura de este documento se seguirá la recomendada: a partir del segundo capítulo hasta el penúltimo se presentarán las fases típicas de un proyecto web, (con la particularidad del capítulo de "Publicación del código", en lugar de la fase típica de "Implantación"). En el último capítulo se encontraran las conclusiones extraídas de la realización de este proyecto así como una opinión personal acerca de la utilización de software libre basada en la experiencia aquí obtenida. Se incluye, así mismo, un anexo con la documentación de usuario del módulo implementado.

1.1. Objetivos

Nuestro proyecto debe permitir la extensión de la funcionalidad de DokuWiki en forma de un nuevo módulo de autenticación para este.

Esto debe permitirnos por un lado la investigación de varios proyectos de código libre desarrollados por terceros para determinar una solución de integración a desarrollar. Todo el proceso debe quedar correctamente documentado (en la presente memoria), así como se deben incluir los anexos correspondientes: presentaciones, documentación de usuario, etc.

Finalmente se deben realizar los procedimientos necesarios para la publicación del código que deberá haber sido implementado en base a la planificación y metodología que se haya previsto en la propia documentación del proyecto.

1.2. Estado del arte y proyectos relacionados

Este proyecto se enmarca principalmente en el contexto de la tecnología de autenticación que proporciona la aplicación SimpleSAMLphp.

SimpleSAMLphp, mediante el protocolo estándar SAML2, proporciona una infraestructura de autenticación distribuida que permite la autenticación en múltiples entornos mediante un proceso de autenticación único. Esto quiere decir que el usuario sólo tiene que introducir sus credenciales una única vez lo que implica al mismo tiempo, que no existe redundancia de datos de autenticación, ni por otra parte, inconsistencia de datos por duplicación de la información de un mismo usuario.

Este procedimiento, aplicado en el contexto web donde actúa nuestro proyecto se conoce como Web Single Sign-On[i].

En cuanto a las dependencias que tendrá nuestro proyecto, tendremos por un lado la aplicación SimpleSAMLphp mencionada, y por otro, DokuWiki, ya que la integración de esta funcionalidad de autenticación se realiza sobre este sistema wiki. También habrá que tener en cuenta las dependencias que tenga este, que son básicamente un servidor web y php 5.1.2.

1.3. DokuWiki

Como en su propio sitio web indica:

> *"**DokuWiki** es un sistema de Wiki[ii] de uso sencillo y compatible con los estándares. Orientado a crear documentación de cualquier tipo dentro de grupos de desarrollo, grupos de trabajo y pequeñas empresas. Su sintaxis[iii] es simple y potente, facilita la creación de textos estructurados, y permite que los archivos generados sean legibles incluso fuera del Wiki. Todos los datos se guardan en archivos de texto*

plano, de tal forma que no se necesita base de datos para su funcionamiento."[iv]

Por su condición de software libre su código fuente es modificable, aunque existe también la posibilidad de extender el software con la implementación de módulos o "plugins"[v]

En cuanto a la instalación de DokuWiki, este provee un script de instalación *install.php* que en el caso de algunos paquetes distribuidos en determinados sistemas ya ha sido ejecutado (aunque se puede volver a ejecutar para cambiar opciones preconfiguradas). Es imprescindible para la instalación en entornos de producción la lectura de la guía de seguridad[vi] donde se detallan los permisos que deben tener los diferentes directorios y ficheros.

Una vez instalado DokuWiki se puede acceder al sistema wiki a través de la dirección *localhost/dokuwiki*, siendo esta la dirección por defecto. Dependiendo de la instalación podremos acceder a la raíz del web de dokuwiki en una ruta u otra (mediante un paquete .deb en Ubuntu esta ruta es /usr/share/dokuwiki).

Por otro lado la configuración del sistema DokuWiki se encontrará en /config o /etc/dokuwiki. DokuWiki hace uso de 2 tipos de fichero de configuración: por un lado los ficheros principales ("main") que vienen ya con DokuWiki en su configuración por defecto. Por otro lado los ficheros locales que son creados por el administrador de DokuWiki.

El fichero de configuración principal más importante es *dokuwiki.php* que debe ser modificado mediante la creación de un fichero *local.php* o *local.protected.php,* de forma que se evitan problemas en las actualizaciones posteriores del sistema (este es el motivo de la existencia de los ficheros de configuración locales). Otros ficheros de configuración importantes son acl.auth.php y users.auth.php, que definen los usuarios y los permisos acl correspondientes. Para ver una lista de los ficheros de configuración existentes: http://www.dokuwiki.org/config#configuration_files

Finalmente, haremos mención de una de las características de DokuWiki que jugará un papel importante en el desarrollo y sobretodo en las pruebas de nuestro proyecto. Se trata del sistema ACL (Access Control List), que permite el control de acceso y permisos a los usuarios en las diferentes páginas del wiki.

El sistema ACL se basa en 3 puntos:

- usuario y/o grupo al que aplicar política de acceso
- página y/o namespace donde aplicar política de acceso
- tipo de permisos a conceder (existen 7 niveles de permisos inclusivos), para los dos puntos anteriores

Para otorgar los permisos se tiene en cuenta que si existen dos reglas con páginas/namespaces diferentes se aplica únicamente la de las páginas o namespace más cercanos. Mientras que si existen reglas distintas para la misma página/namespace (por ejemplo, por que un usuario pertenece a determinado grupo) se aplican los permisos más altos.

1.4. SimpleSAMLphp

Es una aplicación nativa PHP que se encarga de las funciones de autenticación. Principalmente este programa se centra en proveer soporte para el tipo de autenticación conocido como SAML 2.0 tanto como proveedor de servicio como proveedor de indentidad[vii].

SAML2.0 es un estándar de OASIS basado en XML para el intercambio de información de autenticación y autorización entre dominios seguros[viii]. Este estándar básicamente trata el problema de Single Sign-On mediante un navegador web, permitiendo la comunicación entre un Proveedor de servicios y un Proveedor de Identidad.

En la siguiente figura[ix] se puede ver el mecanismo que permite la identificación delegada de los usuarios por parte de la organización que le corresponda a este (haciendo uso de un Proveedor de Identidad), y la aceptación por parte de los diferentes Proveedor de Servicios de su autenticación y autorización (esto es, identificación válida y control de acceso a los recursos correspondientemente):

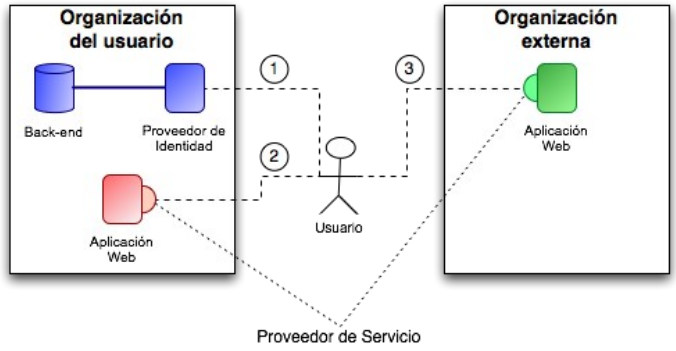


Figura 1. Infraestructura federada de autorización/autenticación

En cuanto a los ficheros de configuración más importantes de SimpleSALMphp los podemos encontrar en el directorio *config*: en *config.php* se encuentra la configuración principal, mientras que en *authsources.php* se puede encontrar información de autenticación de usuarios y sus atributos (incluso se puede configurar la autogeneración de los mismos). También en el directorio *metadata* de deberán configurar los metadatos correspondientes al sistema saml2 en los ficheros del tipo *saml20-**

La forma de integrar este sistema con DokuWiki será mediante el uso del API que nos proporciona SimpleSAMLphp. El API consta de los siguientes métodos definidos:

new SimpleSAML_Auth_Simple(string $authSource)

bool isAuthenticated()

void requireAuth(array $params = array())

void login(array $params = array())

void logout(mixed $params = NULL)

array getAttributes()

mixed getAuthData(string $name)

string getLoginURL(string $returnTo = NULL)

string getLogoutURL(string $returnTo = NULL)

Se pueden encontrar ejemplos de llamadas a estas funciones en
http://simplesamlphp.org/docs/1.8/simplesamlphp-sp-api

1.5. Php (>= 5.1.2)

El siguiente texto que sirve de introducción al lenguaje PHP ha sido extraído de fragmentos del artículo de Wikipedia http://es.wikipedia.org/wiki/PHP sin modificación alguna:

> *"**PHP** es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. […] Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.*
>
> *Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. [...]*
>
> *El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.*

> *Aunque todo en su diseño está orientado a facilitar la creación de sitios webs, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI (Command Line Interface).*
>
> *Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. [...]"*

En nuestro caso, está claro que el uso del lenguaje PHP se enfoca en la creación de una aplicación web dinámica (con orientación a objetos). Aunque como se ha visto el lenguaje PHP tiene una curva de aprendizaje corta por su similitud con lenguajes tan conocidos como C, deberemos contar con una guía para la configuración, el conocimiento de la sintaxis, y el uso correcto de las funciones del lenguaje.

En http://php.net/manual/es/ podemos encontrar la API del lenguaje, definición de todas las funciones, así como toda la documentación mantenida por el PHP Group (que se encarga actualmente de la implementación del lenguaje).

Para la configuración de PHP suele ser suficiente la edición del fichero *php.ini* (que se encuentra en el directorio /etc/php5/apache2 en una instalación con Apache típica en un sistema tipo Debian)

2. Estudio de viabilidad

2.1. Alcance del sistema

El objetivo del presente proyecto es implementar la integración de DokuWiki con simpleSAMLphp de forma que se obtenga un módulo para DokuWiki que permita la autenticación/autorización mediante el protocolo SAML 2.0. Posteriormente se pretende liberar este módulo bajo una licencia libre a elegir.

Para cumplir con estos objetivos será necesario la programación de una clase PHP que realice las llamadas correspondientes al código de la aplicación simpleSAMLphp, (que según definición del propio proyecto es la forma en que se quiere integrar DokuWiki con SAML2). También será necesaria la configuración de Dokuwiki para que integre el código PHP desarrollado. Finalmente, el código php podrá ser "empaquetado" de alguna forma para su fácil instalación (en forma de plugin de DokuWiki por ejemplo).

En cuanto al entorno, para poder utilizar el estándar SAML 2.0 y realizar las pruebas pertinentes sobre el módulo desarrollado es necesaria la configuración de un Proveedor de Identidad (IdP) y un Proveedor de Servicio (SP) proporcionados por el proyecto SimpleSAMLphp.

En cuanto a los aspectos básicos a la hora de evaluar la viabilidad las posibles alternativas al proyecto presentado tenemos:

- **De tipo económico**: el proyecto no debe suponer un coste económico (o ser asumible) al tratarse de un proyecto final de máster.
- **De tipo legal**: el software utilizado debe ser software libre de acuerdo con el máster realizado.
- **De tipo técnico**: la solución a plantear debe tener en cuenta la extensión de un proyecto de este tipo, teniendo en cuenta la utilización de software de código abierto que permita reutilizar componentes evitando la programación de estos desde cero.
- **De tipo operativo**: debe cumplir los requerimientos vistos anteriormente en cuanto a funcionalidad

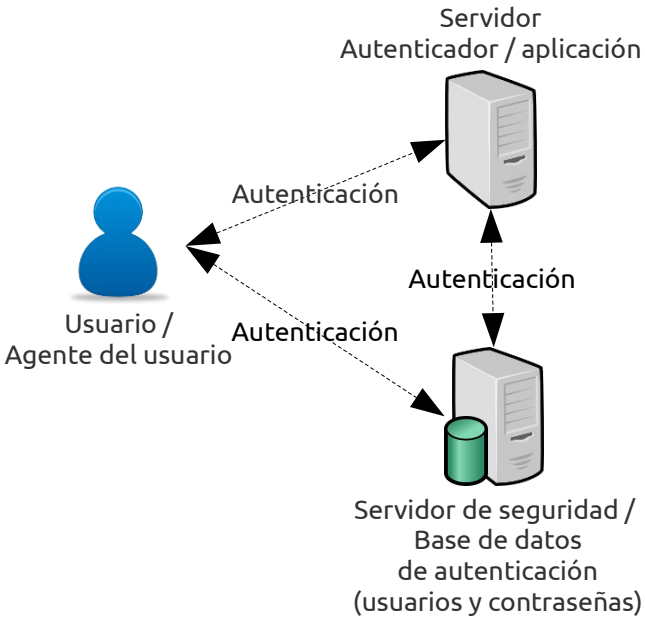
Al tratase de un proyecto de desarrollo no se prevé impacto alguno sobre entorno productivo.

2.2. Estudio de la situación actual

a) Sistemas de autenticación y autorización

Siendo el proyecto tratado un caso de autenticación/autorización federado, es oportuno presentar en primer lugar el estado del arte de los sistemas de autenticación en un caso general para ver donde se sitúa el caso que nos ocupa.

La identificación de un usuario por parte de aplicaciones y sistemas tiene lugar en 2 fases separadas, que son la fase de autenticación y la fase de autorización :



Sistema genérico de autenticación/autorización

Mediante la *autenticación* un servidor autenticador se encarga de comprobar frente a una base de datos, que puede encontrarse en otro servidor, la correcta identificación del usuario. Esta identificación suele producirse mediante el uso de un par nombre de usuario y contraseña.

El usuario, o el agente mediante el cual se comunica este, puede comunicarse bien con el servidor autenticador responsable de proveer los recursos, o en algunos casos con el servidor de seguridad que tiene la información de identificación de forma que sea este quien se encargue del proceso de identificación de forma centralizada sirviendo a los servidores autenticadores o aplicaciones que lo requieran. Este método es el utilizado en los sistemas federados.

En cuanto al proceso de *autorización* será responsabilidad de la aplicación, conforme a las credenciales del usuario obtenidas como consecuencia del proceso de autenticación, permitir diferentes niveles de acceso a los recursos (normalmente según los distintos perfiles de usuario, o grupos, que esta tenga definidos).

Un ejemplo de configuración de autorización por Access Conrol List (ACL) es la que permite DokuWiki mediante su módulo gráfico de configuración:

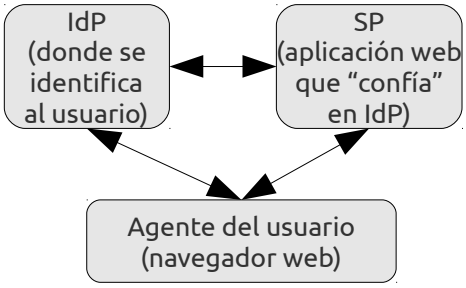
	Page/Namespace	User/Group	Permissions[1)]
#1	*	@ALL	○ None ◉ Read ○ Edit ○ Create ○ Upload ○ Delete
#2	*	@users	○ None ○ Read ◉ Edit ○ Create ○ Upload ○ Delete
#3	*	@staff	○ None ○ Read ○ Edit ○ Create ○ Upload ◉ Delete
#4	private: *	@ALL	◉ None ○ Read ○ Edit ○ Create ○ Upload ○ Delete
#5	private: *	@staff	○ None ○ Read ○ Edit ○ Create ○ Upload ◉ Delete
#6	private:bobspage	bob	○ None ○ Read ○ Edit ○ Create ○ Upload ◉ Delete

Configuración ACL en DokuWiki (GUI)

En cuanto al caso específico de autenticación federada mediante el estándar SAML2 a emplear en este proyecto lo caracterizaremos como una tecnología *SSO* (Single Sign-On), aunque el estándar en si permite otras funciones.

La tecnología SSO evita la duplicación de la información de identificación así como una posible heterogeneidad en esos datos duplicados al mismo tiempo que no requiere por parte del usuario la introducción de sus credenciales para cada sitio con SSO que visite, sino que lo requiere una única vez.

Esto es posible gracias a la centralización de los datos de identificación en el *Proveedor de Identificación* (IdP) que se comunicará tanto con el usuario como con la aplicación / *Proveedor de Servicio* (SP):



El usuario se identifica a través de un agente una única vez (SSO)

Comunicación entre los componentes de un sistema SSO

La inclusión del protocolo XML SAML2 en el proceso de autenticación se puede realizar mediante una aplicación como SimpleSAMLphp (según propuesta del propio proyecto) que simplifica el proceso de autenticación a la utilización de una API y a la configuración de los Proveedores de Identidad y de Servicio vistos en el diagrama anterior.

Estos proveedores o servicios pueden configurarse en una misma máquina o en máquinas separadas, permitiendo una flexibilidad total en la infraestructura de autenticación a implantar.

b) Versiones de los componentes de terceros

Para terminar de caracterizar la situación actual es interesante hacer mención al estado de los diferentes

proyectos que se prevé intervengan en la implantación final, siendo necesario definir las versiones del software sobre el cual se pretende trabajar, así como definir cualquier característica que nos permita realizar un diagnóstico de su situación:

Nombre proyecto	**Versión**	**Notas**
SimpleSAMLphp	1.8	Release actual en la página de DokuWiki (ya que la versión de los repositorios está obsoleta y con problemas)
Php	5.3.3	Versión disponible desde los repositorios del SO perfectamente compatible con el resto de componentes
DokuWiki	0.0.20091225c-8	Versión disponible desde los repositorios del SO, no se han observado problemas de importancia
Ubuntu	10.10	Elección de distribución de uso común en escritorio y como servidor

La elección de un servidor web se podrá ver más adelante en el apartado “Alternativas de la Solución”, donde también se definirá su versión y el porqué de la elección de sus características. También se podrán ver algunos comentarios sobre el porqué de la elección de las versiones de software vistas aquí.

2.3. Definición de requisitos

La definición de los requisitos clave del sistema se clasifica en las siguientes categorías, de forma priorizada según numeración:

Requisitos operativos:
1. Integración de DokuWiki con el servicio SAML2 (mediante SimpleSAMLphp)
2. Creación de un componente o módulo con la funcionalidad del punto anterior que permita su fácil instalación

Requisitos técnicos:
1. Utilización de SimpleSAMLphp como proveedor de servicios SAML2
2. Es conveniente utilizar php para su implementación (ya que es el lenguaje utilizado en el proyecto DokuWiki)
3. Creación de plugin (de adaptarse a las condiciones de nuestro proyecto) conforme a la documentación del proyecto DokuWiki para empaquetar el código desarrollado.

Requisitos legales:
1. Utilización exclusiva de software libre para la realización del proyecto (requisito al ser un proyecto final del Máster Oficial de software libre)
2. Publicación del resultado bajo una licencia libre (según lo acordado en el convenio)

Requisitos económicos:
El coste del proyecto debe ser mínimo teniendo en cuenta que se plantea en el marco de un Proyecto Fin de Máster, que se utiliza software libre, y que se ha considerado que se puede realizar con los recursos propios del alumno al no tener, en principio, requerimientos exigentes en cuanto a hardware ni licencias de software. Es recomendable por tanto que el proyecto se realice en un entorno único (donde se desarrolle y se hagan las pruebas finales).

2.4. Estudio de las licencias libres

Uno de los requisitos que se debe seguir de forma estricta en este proyecto es el uso de software publicado bajo licencias libres o de código abierto. Así mismo el código resultante debe ser publicado bajo una licencia libre.

Para que una licencia sea reconocida como libre o de código abierto debe cumplir con las definiciones de

Free Software Foundation y/o la de la Open Source Initiative[x] (aunque pueden haber algunos desencuentros menores entre ambas). En el caso de la definición de software libre de la FSF queda recogida en 4 libertades básicas:

- *La libertad de ejecutar el programa, para cualquier propósito (libertad 0).*
- *La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.*
- *La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).*
- *La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.*

Una forma rápida de saber si nuestra licencia cumple con estos requisitos es localizarla en un listado donde se clasifican las distintas licencias según la FSF[xi] y OSI:

http://en.wikipedia.org/wiki/Comparison_of_free_software_licences#Approvals

Por otro lado, no todas las licencias libres tienen las mismas características, debemos saber también donde se ubica cada licencia libre respecto al resto para saber que ventajas e inconvenientes puede tener en el desarrollo de nuestro proyecto. Podemos echar un vistazo rápido a estas características, por ejemplo en (aunque es recomendable la lectura completa de las cláusulas de la licencia):

http://en.wikipedia.org/wiki/Comparison_of_free_software_licences#General_comparison

Principalmente podemos ver 2 características básicas que son:

- la posibilidad o no de mezclar proyectos con licencias distintas, dependiendo de si las licencias son compatibles. Básicamente, una licencia con copyleft fuerte no permitirá derivar un trabajo con otra licencia no compatible, mientras que una licencia con copyleft "débil" si lo permitirá manteniendo la compatibilidad únicamente con el código copyleft
- la posibilidad de cambiar la licencia del código libre heredado incluyendo la posibilidad de cerrar el código, lo que se conoce como licencias no copyleft o permisivas.

Sin embargo, la amplia difusión de la licencia GPL, licencia copyleft que permite trabajos derivados exclusivamente bajo los mismos términos de esta licencia, hace que sea de especial relevancia la comprobación de la compatibilidad[xii] de cualquier licencia con GPL y sus versiones, de otra forma el código que se desarrolle con una licencia no compatible puede presentar dificultades para la integración futura en la mayoría de proyectos libres.

Finalmente dentro de las posibilidades que tengamos en la elección de la licencia, deberemos pensar que licencia conviene a nuestro proyecto desde el punto de vista de negocio o de nuestra propia conveniencia. Es decir ¿nos conviene una licencia permisiva o no? ¿que sea compatible con GPL? Para saberlo deberemos haber investigado primero que consecuencias tiene la elección de cada licencia desde todos los puntos de vista: libertad del programador, libertad del usuario, libertad desde el punto de vista empresarial, etc.

2.5. Elección de la licencia libre bajo la que publicar el proyecto

Para la elección de la licencia libre bajo la que queremos publicar el proyecto debemos determinar varias cosas:

- Que proyectos intervienen directamente en la implementación de nuestro módulo
- Que licencias tienen estos proyectos y de que formas se pueden combinar
- Que criterios utilizaremos para la determinación de la licencia

Los proyectos que intervienen de forma más o menos directa en la implementación, dejando a un lado el entorno que utilizamos para el desarrollo y las pruebas, son los siguientes:

- DokuWiki
- SimpleSAMLphp
- PHP

En cuanto a las licencias empleadas:

- DokuWiki se publica bajo una licencia GPL versión 2, copyleft fuerte que solo permite trabajos derivados bajo los mismos términos de licencia

- SimpleSAMLphp se publica bajo una licencia LGPL, que siendo copyleft permite el lincado con otras licencias no libres. En este caso algunas librerías pueden tener otras licencias (se indica BSD como ejemplo). Además, en principio, un trabajo LGPL puede ser derivado en uno GPL

- PHP se publica bajo una licencia permisiva (no copyleft) PHP License, que por causa de una cláusula referida al nombre de los trabajos derivados es incompatible con la licencia GPL.

Para conocer de que manera podemos combinar las licencias de los proyectos en primer lugar debemos tener en cuenta que clase de proyecto estamos implementando. Al tratarse de la implementación de un *plugin* para DokuWiki debemos adaptarnos a los términos de licencia de este proyecto, teniendo en cuenta lo que dice la licencia GPL respecto a los plugins[xiii]. Como nuestro plugin no se ejecuta como un programa independiente, mediante exec o fork, deberá publicarse bajo una licencia compatible con GPL (una de sus versiones).

Por otro lado nuestro proyecto utiliza la librería de SimpleSAMLphp, que se carga en tiempo de ejecución para poder utilizar su API en nuestro código. Al tratarse de una licencia LGPL compatible con la licencia GPL no deberemos tener problema[xiv] en licenciar nuestro proyecto bajo esta última licencia.

No se considera el uso del lenguaje PHP como determinante en la licencia del proyecto, aunque ambos proyectos están implementados en este lenguaje. En el caso de los lenguajes interpretados el código de ese lenguaje no se diferencia en nada de cualquier otro tipo de datos[xv]. Será el intérprete de ese lenguaje el que estará sujeto a la licencia correspondiente, PHP License en este caso.

En este caso nuestra libertad de elección de licencia final, si queremos usar los componentes indicados, se limitaría a la elección entre las diferentes versiones de la licencia GPL. Elegiremos entonces la licencia GPLv2 para publicar nuestro plugin por tratarse de la misma licencia y versión utilizadas en el proyecto DokuWiki, ya que en definitiva nuestro trabajo es una extensión de este proyecto.

2.6. Estudio de las alternativas de solución

Debido a la orientación técnica del presente proyecto, en que los objetivos del mismo delimitan claramente el abanico de soluciones a utilizar (el protocolo SAML2, SimpleSAMLphp, DokuWiki y Php), en el presente apartado se planteará únicamente que aplicaciones deben dar soporte a las aplicaciones principales que ya han sido mencionadas en los apartados introductorios.

Cabe reseñar brevemente la existencia de otros protocolos de autenticación similares a SAML2, entre los que se encuentran Cosign, JOSSO o Shibboleth, así como de otras aplicaciones Wiki, que nos llevarían probablemente al uso de otras tecnologías que no se han visto aquí. Por ejemplo Java en lugar de PHP y un servidor web acorde a ese lenguaje de programación.

Las alternativas entre las que podremos escoger en este punto son: sistema operativo, servidor web y aplicaciones de desarrollo y documentación.

En la elección de sistema operativo, concretamente, se deben considerar los siguientes puntos:

- necesidad de que el sistema elegido permita la realización de todas las tareas del proyecto. Esto incluye tanto las tareas propias de desarrollo y documentación así como la implantación y pruebas en un entorno de servidor
- compatibilidad con las soluciones ya definidas (Php, etc.)
- necesidad de que sea libre legalmente y no constituya coste económico alguno
- repositorios con aplicaciones periódicamente actualizadas

En cuanto a la compatibilidad con las soluciones basadas en Php no constituye ningún problema en ningún

SO debido a la característica multiplataforma de este lenguaje. En cambio la necesidad de un sistema libre y sin coste, con unos repositorios relativamente actualizados, nos decanta por la rama GNU/Linux de sistemas operativos, existiendo todavía múltiples alternativas:

- Debian GNU/Linux
- Ubuntu
- Fedora
- Opensuse

No existen grandes diferencias entre estos sistemas en cuanto a funcionalidad así que simplemente elegiremos el que tiene mayor aceptación actualmente, Ubuntu, en su versión estable 10.10, que es un derivado de Debian GNU/Linux.

La elección del servidor web, además de ser libre debe responder básicamente a la compatibilidad de este con Php y Openssl (que podría ser requerido en el proyecto). Debido a que todos los servidores principales tienen estas características, decidiremos usar el servidor del que disponemos más referencias en la documentación de DokuWiki y SimpleSAMLphp, que es Apache2. Este servidor libre es el que tiene mayor cuota de mercado muy por encima de su competidor propietario IIS, según Netcraft[xvi]:

Desarrollador	**Nombre**	**Páginas alojadas**	**Porcentaje**
Apache	*Apache*	179,720,332	60.31%
Microsoft	*IIS*	57,644,692	19.34%
Igor Sysoev	*nginx*	22,806,060	7.65%
Google	*GWS*	15,161,530	5.09%
Lighttpd	*lighttpd*	1,796,471	0.60%

En cuanto a las aplicaciones de documentación y desarrollo, basaremos su elección en su licencia que deberá ser libre. A continuación se muestra una lista de las aplicaciones escogidas que podrán ser descargadas directamente de los repositorios del sistema:

- *Vim*, editor de texto y entorno IDE, posibilidad de usar plugin para php
- *OpenOffice*, procesador de textos, gráficos y presentaciones de diapositivas
- *ArgoUML*, modelado UML
- *Umbrello*, modelado UML
- *Planner*, gestión temporal del proyecto
- *Gimp*, edición de imágenes

2.7. Metodología

a) Ciclo de vida del proyecto

La metodología que se debe seguir en todo proyecto de software debe permitir refinar la solución que se pretende obtener mediante fases que persiguen diferentes objetivos cada una de ellas. En nuestro caso se seguirá el ciclo de vida clásico de un proyecto que consiste en la puesta en marcha de cada una de estas fases de forma ordenada al haber terminado la fase inmediatamente anterior.

En la figura inferior se pueden ver estas fases, que en la mayoría de proyectos suelen terminar con la fase de implantación seguidas de la fase de mantenimiento. En nuestro caso, la última fase se ha caracterizado como *Publicación/Mantenimiento* lo que sugiere que el proyecto no persigue el desarrollo de un proyecto para su implantación en un entorno productivo sino simplemente la publicación mediante una licencia libre para que este código pueda ser implantado en aquel entorno que sea considerado conveniente. También

contempla la posibilidad de mantenimiento posterior gracias a la disponibilidad pública del código y documentación anexa.

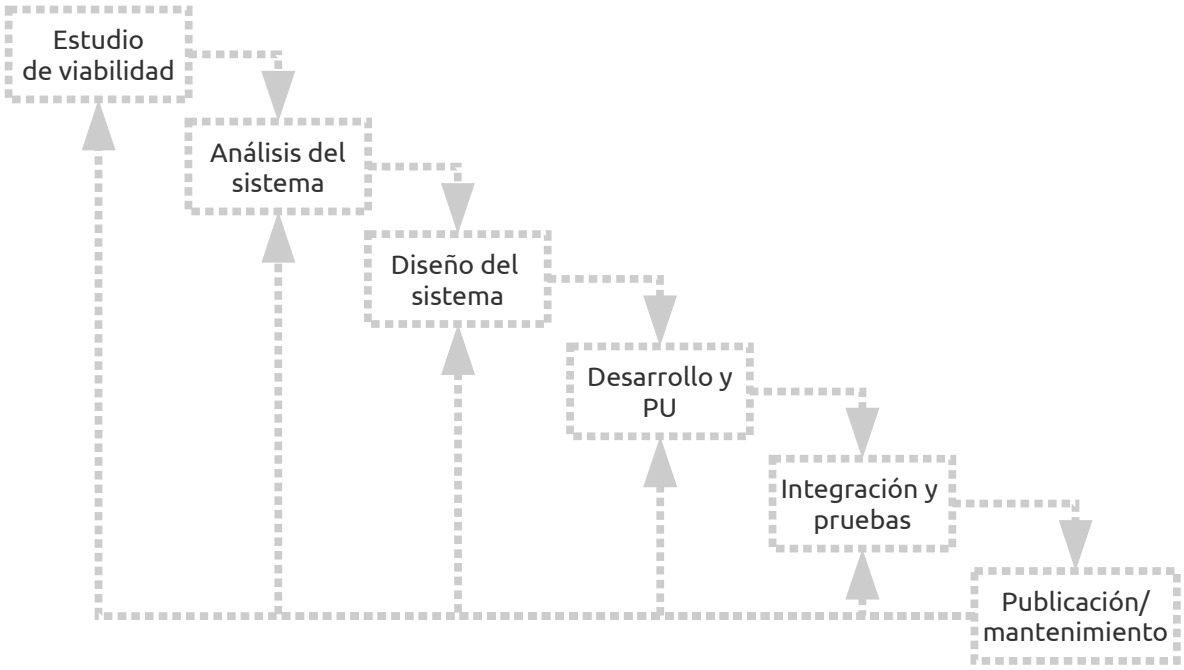


Diagrama del ciclo de vida clásico o "en cascada"

b) Estándares y estilo

En cuanto al seguimiento de estándares, en la fase de análisis y diseño se hará uso del lenguaje de modelado UML que permite la definición de software orientado a objetos. En la fase de diseño/desarrollo se seguirán las APIs que proporcione cada proyecto en su documentación (PHP, SimpleSAMLphp y DokuWiki).

El estilo del código a implementar deberá contemplar los comentarios pertinentes para la "fácil" comprensión del mismo, pero especialmente se deberá hacer una descripción de cada una de las funciones o clases de forma que quede claro cual es su funcionalidad sin necesidad de ver el código. La clase/s a implementar deberán situarse en un fichero a parte.

c) Entregas

Al mismo tiempo que se avance con las diferentes fases del proyecto se entregarán las PAC correspondientes a la asignatura TFM e informes del avance del proyecto, así como se presentarán dudas al consultor externo (asignatura PFM).

Finalmente se realizará la entrega final que incluirá además del código, una memoria en la que conste la documentación de cada una de las fases del proyecto extraída principalmente de las PAC de la asignatura TFM a las que se han hecho referencia, así como una presentación y un vídeo ilustrativo del funcionamiento correcto de la implementación.

2.8. Valoración de costes económicos y riesgos asociados a la solución propuesta

a) Valoración de costes

Para acotar los costes económicos del proyecto, resultado de la estimación de costes y beneficios, se

deben tener en cuenta tanto los costes tangibles (precio de licencias), como intangibles (horas de dedicación en la implementación).

Coste económico en software:

SO: Ubuntu 10.10
Web: Apache 2.0 y PHP, Dokuwiki, SimpleSAMLphp, OpenSSL
Documentación: OpenOffice, Planner, Gimp
Desarrollo: ArgoUML, Vim, Firefox

Todo el software utilizado tiene licencias libres sin coste: 0€

Coste del hardware:

Se utiliza PC propio para documentación, desarrollo e implantación

El coste es 0€

Coste horas trabajadas:

184h según la planificación actual

En el contexto actual son 0€, en un contexto hipotético se pude considerar un sueldo de 10 €/hora, considerando una media de los sueldos de los diferentes perfiles que intervienen en desarrollo, documentación de usuario e implantación, lo que daría un total final de 1840 €

Para determinar la viabilidad económica, por lo tanto los beneficios reportados en una situación en un entorno "laboral" (no un proyecto universitario) deberían compensar los 1840€ que se corresponderían totalmente con el coste del salario de los recursos humanos.

b) Riesgos asociados a la solución de integración

Al tratarse de una solución de integración de dos proyectos los riesgos potenciales que se pueden identificar son los de posibles cambios en estos proyectos que "rompieran" nuestro componente de integración.

La solución de contingencia más adecuada es asegurar que la elección de las versiones elegida de ambos proyectos sean lo suficientemente estables como para que en caso de realizarse pequeños reajustes en las interfaces de los mismos, nuestro proyecto se pueda adaptar fácilmente, de forma que el esfuerzo puesto con anterioridad no sea inútil. Por ejemplo, un cambio en el formato de los plugins de DokuWiki en nuevas versiones de este software se debería poder solucionar de forma rápida y sencilla, es decir el impacto de un problema de este tipo sería asumible

2.9. Conclusión del estudio de viabilidad

La conclusión del estudio de viabilidad en cuanto a la solución planteada en los apartados anteriores se apoya en los requisitos vistos desde diferentes ángulos (económico, técnico, legal y operativo)

Sin embargo, también debemos situar el actual proyecto en su contexto de proyecto universitario, para poder concluir su idoneidad. Está es un análisis general en el contexto del PFM:

- **Roles a desempeñar**: desarrollo, administración, documentación y gestión de proyectos
- **Áreas**: web y seguridad, adecuadas al desarrollo personal del máster según las asignaturas cursadas
- **Adecuación al convenio**: la planificación y la carga de trabajo en horas que establece el convenio se corresponden (185h)

- **Software utilizado**: se encuentran dentro del contexto del máster al tratarse todas ellas de software libre
- **El proyecto es software libre**, es adecuado al PFM.

Teniendo en cuenta esto, y que el proyecto es también viable en las vertientes económica, técnica, legal y operativa, ya que tiene un coste cero, es técnicamente factible, cumple con los requisitos legales y la funcionalidad establecida como se ha visto a lo largo de este estudio.

3. Análisis del sistema

3.1. Definición del sistema

a) Requisitos exactos

Partiendo de los requisitos expuestos en el estudio de viabilidad sabemos que el módulo a implementar debe comunicar DokuWiki con el Proveedor de Servicios SAML2 mediante SimpleSAMLphp. Para hacerlo se deben cumplir los siguientes requisitos:

- La comunicación entre el Proveedor de Servicios SAML2 y el software DokuWiki debe realizarse mediante la *API simpleSAMLphp del SP*[xvii]
- La integración del código con las llamadas a la API debe realizarse desde una nueva clase que extienda la clase base *auth_basic* (en basic.class.php) de DokuWiki
- La nueva clase debe implementar, al menos, la función *trustExternal* para proporcionar el servicio de autenticación externo SSO de SimpleSAMLphp al sistema DokuWiki
- Se debe configurar DokuWiki para que cargue las librería de SimpleSAMLphp así como utilice la autenticación de la nueva clase creada
- Se debe empaquetar el código de la nueva clase, así como las posibles configuraciones. Cabe la posibilidad de que esto se haga mediante el mecanismo de plugin para DokuWiki siguiendo el estándar de la documentación[xviii]
- El módulo implementado debe ser configurable para que este se pueda adaptar a diferentes configuraciones de entornos SimpleSAMLphp

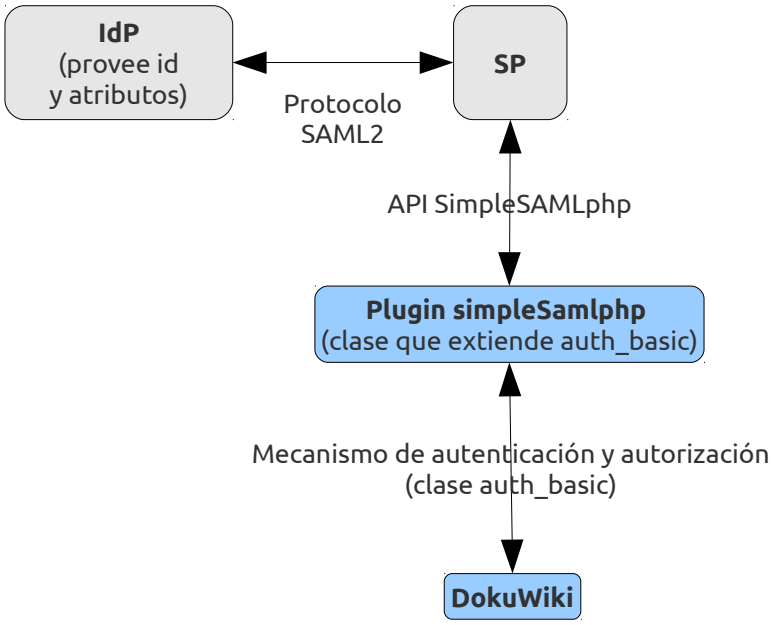


Diagrama de los componentes del sistema

b) Entorno tecnológico

En la siguiente figura podemos ver un perfil SSO tal y como aparece en la documentación de OASIS del estándar SAML2[xix]. La figura corresponde a uno de los múltiples posibles perfiles[xx] que cumplen el estándar en el caso de SSO y nos sirve para situar el componente que debemos desarrollar en un contexto más amplio.

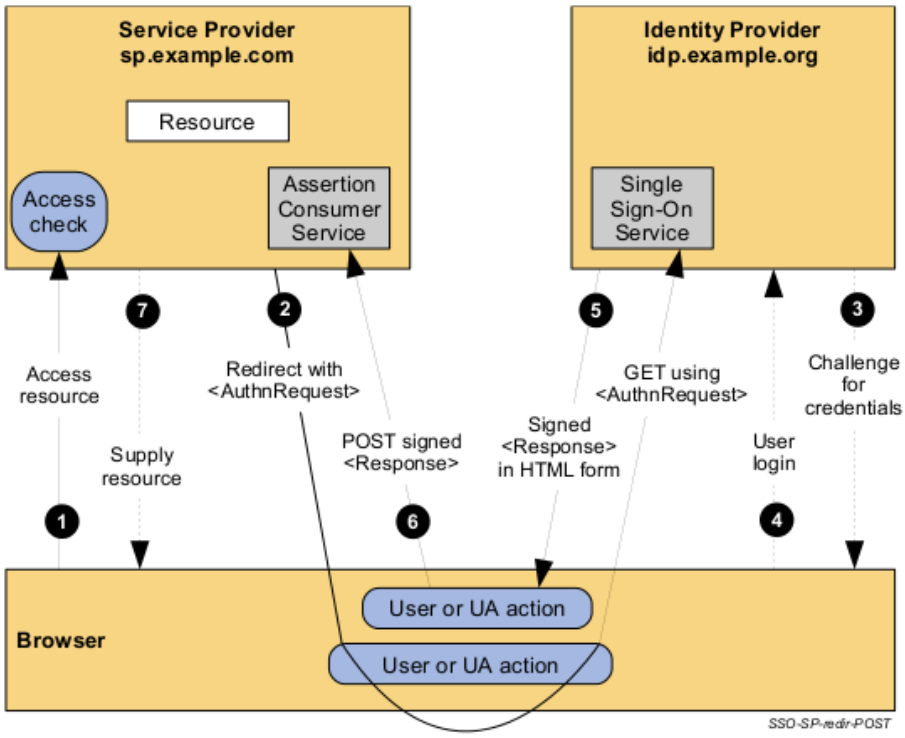


Diagrama perfil SSO iniciado en SP que hace uso de redireccionamiento y POST

Concretamente el perfil SSO en cuestión incia el proceso de SSO en el SP, siendo el usuario quien intenta acceder a un recurso de este (1 en el diagrama anterior). En el caso de no tener una sesión abierta el SP hace una redirección hacia el IdP (2) que le envía un formulario al usuario donde puede autenticarse (3), posteriormente en caso de éxito (4), envía las credenciales del usuario (atributos) al SP automáticamente mediante HTTP POST (5 y 6).

Sin tener en cuenta la forma en que se comunican SP e IdP - en este caso lo hacen indirectamente a través del agente del usuario mediante una redirección y HTTP POST - cualquier perfil SSO deberá tener en común el primer paso, donde el agente del usuario hace una petición sobre un recurso. Este es en realidad el único punto donde la clase a implementar tiene lugar para su actuación, ya que el resto del código se ejecuta en los Proveedores SAML2 o en el agente del usuario, interviniendo este último solamente para proporcionar un nombre de usuario y contraseña.

Por otro lado existe la necesidad de configuración del entorno SAML2 con unos metadatos encargados de otorgar confianza mutua al SP e IdP que intercambian datos de SSO.

Tanto los mensajes intercambiados como los metadatos siguen el estándar definido por SAML2, XML, sin embargo la capa SimpleSAMLphp nos permite abstraernos del XML para trabajar, en el caso de los metadatos y otras configuraciones, directamente con php.

Para la comprobación del funcionamiento correcto de nuestro SP se puede hacer uso de Feide OpenIdp[xxi]. Sin embargo en nuestro caso será necesaria la configuración de un IdP propio, en la misma máquina, que nos sirva la información de autenticación para el acceso a los recursos de DokuWiki.

Hay que tener en cuenta que se pueden usar distintos modos de almacenamiento para la información de autenticación[xxii] del IdP. Podemos destacar el modo “UserPass” (fichero php editado manualmente con array con los usuarios) y “SQL” (base de datos). Cualquiera de estos 2 métodos es válido, siendo el segundo caso más costoso, al no contar el sistema DokuWiki con una base de datos por defecto para ser conectada a nuestro IdP. Deberíamos, por tanto, crear unas tablas exclusivamente para proporcionar la información de autenticación al IdP, lo cual parece innecesario si tenemos en cuenta que no es nuestra intención prepara un entorno de pre-producción, sino un entorno de desarrollo y pruebas.

Por otro lado la configuración ACL seguirá estando del lado de la aplicación ya que la intervención de este sistema de control, que forma parte de DokuWiki, otorga privilegios confiando en que la autenticación se ha hecho de forma correcta.

En cuanto a la definición de los usuarios del sistema hay que tener en cuenta que estos siempre interactuaran con el sistema de la misma manera (mediante el agente anteriormente referido), sin embargo será necesario para poder realizar las pruebas pertinentes que se definan como pertenecientes a diferentes “grupos” de forma que se pueda comprobar como afecta esto a la respuesta que se obtenga a la petición de recursos, (7 en la figura anterior), una vez se apliquen las políticas ACL.

Destacar por último la importancia del uso del estándar OpenSSL y HTTPS en entornos de producción, según señala el proyecto DokuWiki. En nuestro caso como esto solo afecto a nuestro entorno de desarrollo se puede valorar el uso de una instalación de SimpleSAMLphp corriendo simplemente sobre HTTP.

3.2. Establecimiento de requerimientos

El establecimiento de requerimientos permite tener una visión organizada de los requisitos anteriormente descritos. Cada requisito establecido detalla características bien de tipo funcional, de rendimiento, de seguridad, de implantación o de disponibilidad.

Para la definición clara y concisa de estos requisitos a su vez se elaboran los casos de uso, que definen las interacciones del propio sistema con sus usuarios. La revisión del conjunto de casos de uso debe permitir la elaboración de un diagrama UML de casos de uso que muestra las relaciones entre los usuarios y los distintos casos de uso del sistema.

La comparación de los requerimientos establecidos en este apartado con el diseño elaborado posteriormente nos permitirá comprobar la validez de este.

Los requerimientos detectados son:

- Cualquier usuario que no ha hecho login en el IdP correspondiente (ni a través de DokuWiki ni por otro medio), debe obtener un formulario para el login en el IdP. El usuario debe introducir nombre y contraseña, y el IdP debe autenticar (o no) al usuario en el servicio final enviando a este los datos asociados al usuario en cuestión.

- Cualquier usuario que se ha autenticado en el IdP, a través de DokuWiki o otro medio, debe poder acceder a los recursos de DokuWiki que este le permita según el ACL de DokuWiki y el grupo al que pertenezca que se deberá haber obtenido previamente del IdP, al estar el usuario ya autenticado.

- Cualquier usuario que se ha logueado en el IdP, a través de DokuWiki o otro medio, debe poder salir de su sesión actual en el IdP mediante el procedimiento estándar de log out de DokuWiki, saliendo al mismo tiempo de cualquier otro servicio asociado al mismo IdP.

- Un administrador de DokuWiki deberá poder instalar el módulo de autenticación siguiendo las instrucciones provistas en el paquete del módulo de forma manual. No se considera conveniente la automatización de la instalación mediante el uso de un plugin de DokuWiki ya que de todas formas se necesitan permisos de administración sobre el sistema de ficheros. Cada distribución puede requerir la instalación en una ruta distinta y la instalación de los llamados plugins de DokuWiki se

suele realizar en el directorio /lib/plugins que sí debe tener dados permisos para la instalación de plugins no siendo este nuestro caso.

Vemos como a partir de los requerimientos anteriores podemos extraer la relación de un usuario interactuando con una parte del sistema conformando los casos de uso siguientes:

Actor principal	***Caso de uso / requerimiento***	***Tipo de requerimiento***
Usuario no autenticado en IdP	Log in usuario DokuWiki	funcional
Usuario autenticado en IdP	Acceso a recursos sin login	funcional
Usuario autenticado en IdP	Log out usuario	funcional

De donde se puede inferir el diagrama de casos de uso siguiente:

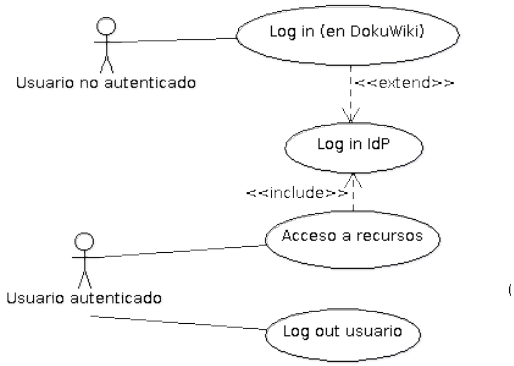


Diagrama de casos de uso

3.3. Fichas de casos de uso

Caso de Uso	Log in usuario DokuWiki

Actor:	Usuario no autenticado
Descripción:	Procedimiento para la autenticación de un usuario no identificado en DokuWiki
Precondiciones:	1. Usuario no identificado en IdP
Postcondiciones:	1. El usuario debe estar autenticado, de lo contrario recibirá un mensaje conforme a que los datos introducidos no son válidos o que ha ocurrido un error recuperando los datos (los datos suministrados por el IdP no son válidos).
Flujo normal:	1. El usuario hace click en un enlace de DokuWiki para iniciar el proceso de log in 2. Se presenta al usuario un formulario para que introduzca usuario y contraseña (por parte del IdP al que ha sido redirigido) 3. El usuario introduce un nombre y una contraseña 4. IdP comprueba la validez de los datos enviados por el usuario y en caso positivo responde con la autorización de autenticación para el sistema proveedor y DokuWiki, así como con información asociada al usuario, principalmente grupo al que pertenece

	5. Se autentica al usuario en DokuWiki, con los permisos correspondientes a sus credenciales (nombre de usuario y nombre de grupo/s)
Flujos alternativos:	4. Si el IdP comprueba que los datos no son válidos vuelve a mostrar un formulario para que el usuario introduzca los datos correctos advirtiendo del error en los datos provistos previamente 4. Si el IdP no devuelve datos válidos para DokuWiki se muestra un mensaje de error y un enlace para poder hacer logout en el IdP

Caso de Uso	Acceso a recursos sin login

Actor:	Usuario autenticado en IdP
Descripción:	Acceso a los recursos de DokuWiki mediante ACL
Precondiciones:	1. Usuario identificado en IdP
Postcondiciones:	1. El usuario debe acceder a los recursos sin tener que introducir ninguna información de autenticación. Si los datos del IdP no son válidos se informa al usuario y se prorciona un enlace para cerrar sesión en el IdP
Flujo normal:	1. El usuario accede a cualquier URL de DokuWiki 2. Si se encontraba identificado en DokuWiki accede a los recursos disponibles conforme a sus credenciales
Flujos alternativos:	1. DokuWiki comprueba que, aunque el usuario no se había autenticado directamente en DokuWiki, sí había sido autenticado por IdP anteriormente 2. Se autentica al usuario en DokuWiki, accediendo a los recursos conforme a sus credenciales (nombre de usuario y nombre de grupo)

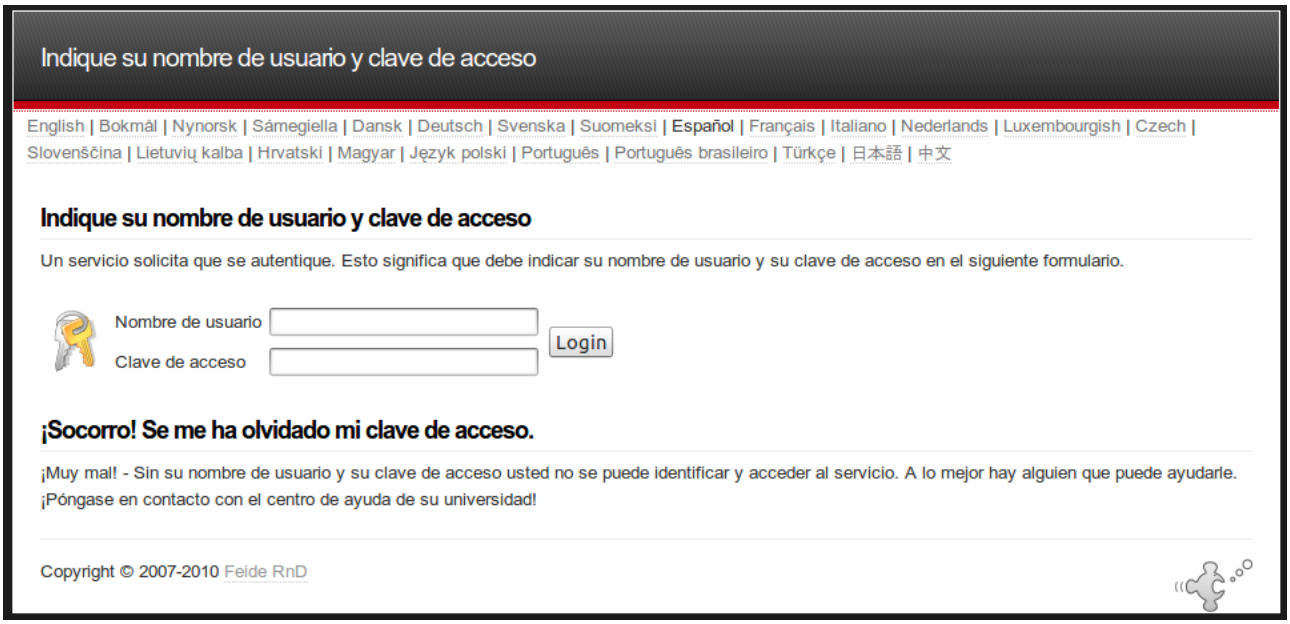
Caso de Uso	Log out usuario

Actor:	Usuario autenticado en IdP
Descripción:	Se cierra la sesión de autenticación existente en IdP y DokuWiki
Precondiciones:	1. Usuario identificado en IdP
Postcondiciones:	1. El usuario no debe poder acceder a los recursos de DokuWiki ni a los de ningún otro servicio asociado al IdP
Flujo normal:	1. El usuario hace click en un enlace de DokuWiki (logout) 2. Se comunica al IdP el fin de la sesión de autenticación 3. DokuWiki y IdP cierran la sesión del usuario

3.4. Definición de las interfaces de usuario

Para la integración de DokuWiki con SimpleSAMLphp no se requiere la implementación de ninguna interfaz de usuario nueva ya que la única interacción que el nuevo sistema requiere con el usuario - la petición de un nombre de usuario y una contraseña - es realizada ya a través de los servicios del IdP, es decir de la aplicación SimpleSAMLphp.

La interfaz proporcionada por SimpleSAMLphp puede ser personalizada para incluir alguna imagen que distinga el servidor de autenticación en cuestión IdP. Esto queda fuera del alcance del proyecto ya que no forma parte del módulo de autenticación.



Formulario provisto por defecto para la introducción de datos de usuario

3.5. Planes de pruebas

En la actual fase de análisis se define el marco de las pruebas a realizar (y su tipología), será sin embargo en la fase de diseño donde se detallarán las pruebas hasta el nivel de qué entradas y qué salidas requiere cada prueba realizada.

Para la aceptación del sistema se pueden llegar a requerir pruebas de las siguientes tipologías:

- pruebas unitarias: comprueban el funcionamiento de componentes individuales
- pruebas de integración: comprueban el funcionamiento de módulos de forma combinada
- pruebas de implantación: comprobaciones sobre el entorno de producción
- pruebas de aceptación: pruebas que realizan los usuarios del sistema con el fin de validar definitivamente el sistema (funcionalidad, seguridad, rendimiento, disponibilidad...)

Las pruebas de implantación en este caso no serán realizadas, ya que el proyecto no requiere implantación. Las pruebas de aceptación deberán ser realizadas externamente por quien deba validar el correcto funcionamiento del proyecto.

Así mismo se deberán definir los requerimientos (alcance) y requisitos para su realización. El alcance de las pruebas deberá describir para qué sirve la prueba, qué usuarios están implicados y qué criterios se exigen para la aceptación de la prueba. Mientras los requisitos de las pruebas harán referencia a qué hardware y software será necesario y como deberá estar configurado, etc.

Se pueden determinar los siguientes usuarios o actores en las pruebas:

- usuario no autenticado en IdP
- usuario autenticado en IdP
- administrador de DokuWiki

En cuanto a los criterios funcionales para la redacción y aceptación de las pruebas se deberán seguir los casos de uso de este análisis, dando lugar a diferentes pruebas de integración. Por tanto los planes de pruebas se corresponderán con los siguientes escenarios:

- log in usuario DokuWiki
- acceso a recursos ACL
- log out usuario

Por otro lado, las pruebas unitarias se corresponderán mayormente con la comprobación de la correcta ejecución del código implementado, también conocido como pruebas de caja blanca.

Los requisitos de hardware y software que se deberán cumplir para la realización de las pruebas serán los mismos que para el desarrollo, ya que se realizaran sobre el mismo entorno.

Se deberá realizar una copia de seguridad con el estado inicial de los datos de pruebas a fin de evitar el reproceso manual de estos datos cada vez que se deba repetir una prueba.

4. Diseño del sistema

4.1. Arquitectura

Los modelos y especificaciones a obtener en la presente fase consistirán de la elaboración de una serie de diagramas que definan diferentes perspectivas de los componentes.

La arquitectura del sistema a implementar consistirá en la suma de la arquitectura conceptual y la arquitectura lógica (encargada de definir las comunicaciones entre componentes). Como en el caso que nos ocupa estamos desarrollando un componente para un sistema mayor, la definición de la arquitectura conceptual que define los grandes bloques del sistema nos debe situar en el contexto global. Estos grandes bloques son: El sistema wiki *DokuWiki,* el módulo de autenticación SAML2, el *Service Provider* y el *Identity Provider provistos por SimpleSAMLphp.*

Por otro lado, podrían existir subcomponentes en estos grandes bloques, caracterizados cada uno de estos por funcionalidad común, acceso a datos comunes, etc.

Analizando los componentes anteriores, vemos que podemos considerar al Identity Provider como la composición de un servidor de identidad, y de un sistema de almacenamiento de datos de autenticación (aunque en nuestro caso este sistema no sea externo sino simplemente un fichero de configuración).

Más abajo, en el apartado “Diagramas UML”, se pueden ver los diagramas de componentes y de interfaces que se corresponden con la mencionadas arquitectura conceptual y arquitectura lógica.

4.2. Normas y estándares para el diseño

Podemos clasificar las normas y estándares a seguir por el proyecto en 3 puntos:

- un formato para documentar el proceso de diseño
- una notación para los diagramas
- definición de idioma y estilo

a) Formato de la documentación

El formato de la documentación de diseño debe ser un formato libre, que permita la consulta y/o edición de la misma por diferentes grupos de trabajo sin limitar su accesibilidad. Por ello se elige el formato OpenDocument, así como el formato PDF al cual es posible la exportación desde OpenDocument para visualizar la documentación en modo de sólo lectura.

Este documento deberá incluir los cambios que se puedan producir respecto al diseño original dejando constancia de que cambios se realizan, en que fecha y quien es el responsable del mismo.

De la misma forma que el formato debe ser libre también debe tenerse en consideración la inclusión de diagramas u otros elementos en la documentación que puedan ser modificables fácilmente. Para ello estos deberán ser construidos también con herramientas libres que aseguren la posibilidad de modificación así

como la accesibilidad desde cualquier plataforma libre. Será por tanto aconsejable adjuntar junto con la documentación los ficheros de gráficos susceptibles de modificación (siempre que estos no sean modificables dentro del documento)

b) Notación de los diagramas

La notación de los diagramas será la notación UML que permite representar el paradigma de la orientación objetos para su análisis[xxiii]. Se considerará conveniente la confección de diagramas de componentes/despliegue así como diagramas de clases. Atendiendo a todo esto, existen varios programas libres que permiten trabajar con esta notación, entre ellos ArgoUML y Umbrello. Habiendo probado ambas aplicaciones se han detectado algunas limitaciones en ArgoUML (en lo que atañe al diseño de interfaces) por lo que se ha considerado más conveniente la utilización del software Umbrello.

c) Definición de idioma

La definición del idioma a utilizar se realiza teniendo en cuenta 2 criterios:

- la necesidad de redactar el conjunto de la documentación en un mismo idioma (es decir, la memoria del proyecto). Desde el inicio se eligió el castellano.
- en los proyectos de software libre se considera una buena práctica la inclusión de comentarios en el código en inglés, al tener esta lengua una mayor aceptación internacional. Además siendo nuestro proyecto una extensión de otro (un plugin), es una buena opción utilizar el mismo idioma que se utiliza en el resto del código.

Por lo tanto, mientras que en la documentación se utiliza el castellano, en los comentarios se utiliza el inglés. También se utilizará el inglés por tanto en el nombre de funciones, clases y variables, así como en los ficheros de texto que se empaqueten junto con el código, que proporcionen instrucciones básicas de instalación (INSTALL.TXT).

d) Definición de estilo

En la definición del estilo de codificación se puede tomar en consideración cual es el criterio seguido en el resto del código perteneciente al proyecto DokuWiki, como ejemplo podemos tomar algunos fragmentos de la clase de donde debe derivarse la clase a implementar:

```
<?php

class auth_basic {

  var $success = true;
[...]
  function canDo($cap) {
    switch($cap){
      case 'Profile':
        // can at least one of the user's properties be changed?
        return ( $this->cando['modPass']  ||
                 $this->cando['modName']  ||
                 $this->cando['modMail'] );
        break;
[...]
      default:
        // print a helping message for developers
        if(!isset($this->cando[$cap])){
          msg("Check for unknown capability '$cap' - Do you use an outdated Plugin?",-1);
        }
        return $this->cando[$cap];
    }
  }
```

[...]
}

Del código anterior podemos extraer algunos estándares de codificación:

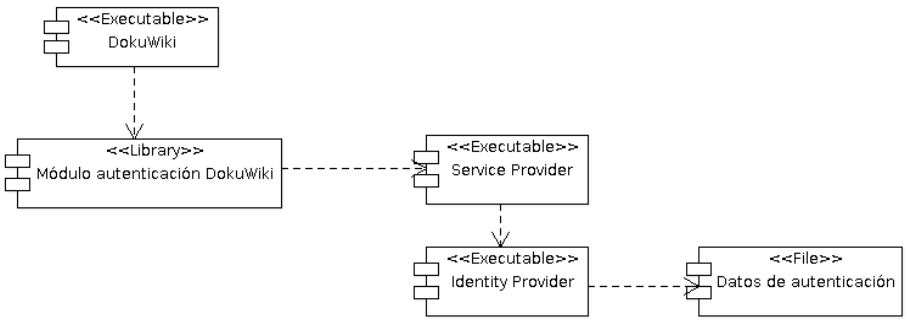
- Usaremos <?php en lugar de <?
- Usaremos 2 espacios en lugar del tabulador para indentar (programas como Vi convierten el carácter tabulador en espacios, si estamos acostumbrados a usarlo)
- Para definir las funciones y las clases la llave de apertura '{' estará en la misma línea que el nombre de la función/clase
- Los condicionales tendrán la llave de apertura '{' también en la misma línea, no en una línea nueva. Ejemplo: if (foo == bar) {
- En una condición else, la llave de cierre, else, y llave de apertura iran en la misma línea.

  Ejemplo: } else {
- Las funciones tendrán la siguiente notación: nombreFunción
- Las variables tendrán la siguiente notación: nombreVariable
- Las clases tendrán un nombre tipo: nombreclase.class.php
- Se dejaran espacios entre paréntesis y operadores: (par), foo == bar
- Se dejaran espacios en las asignaciones, delante y detrás: $foo = bar

4.3. Diagramas UML

A continuación se presentan los diagramas UML de componentes y de interfaces que corresponden a la arquitectura conceptual y arquitectura lógica, así como al modelo de programación orientada a objetos representado por un diagrama de clases.

a) Diagrama de componentes

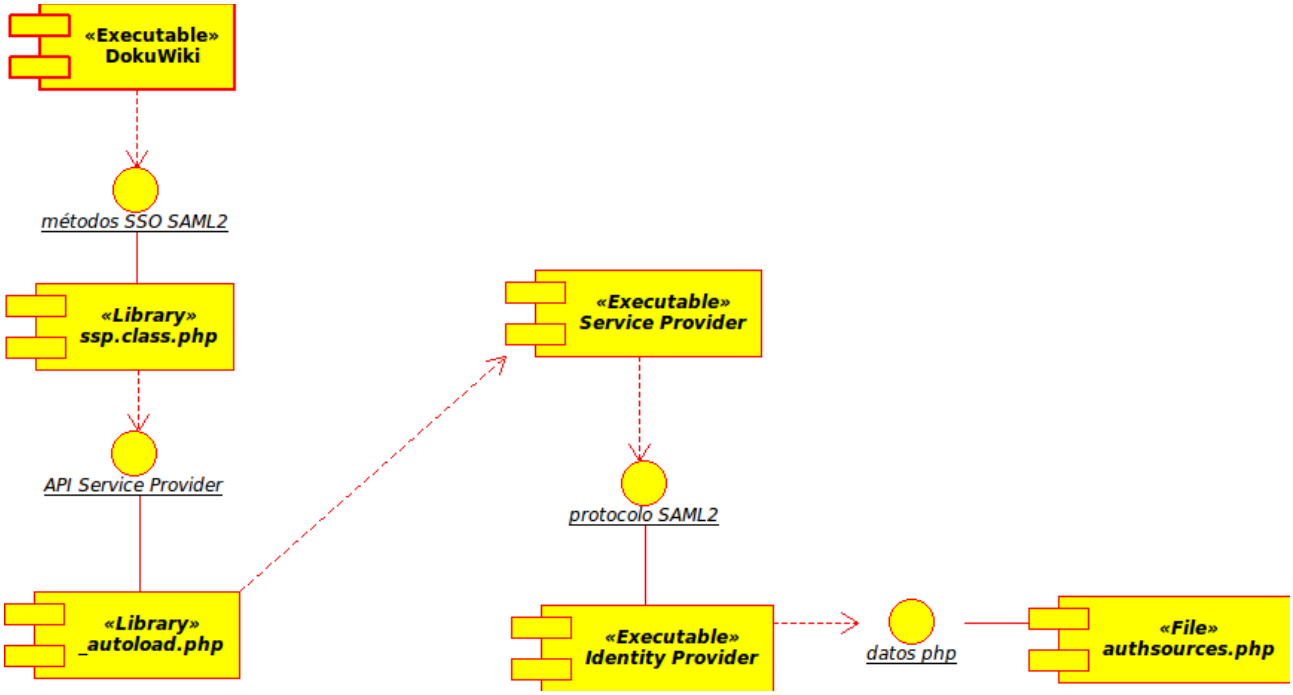


En este diagrama de componentes se ha separado el componente Identity Provider en dos partes: el Servidor IdP, y el Servidor de datos de autenticación. De esta forma se ha dejado claro que el Identity Provider puede hacer referencia a información de autenticación externa (almacenada en la base de datos de uno de los sistemas a los que permita el acceso por ejemplo).

Además se han caracterizado las 2 instancias Service Provider y Identity Provider como 2 procesos que se

ejecutan por separado (y que pueden ejecutarse por tanto en nodos distintos). El módulo de autenticación de DokuWiki por su parte se caracteriza como una simple librería (estereotipo “Library”), ya que no hay que olvidar que es simplemente una clase y las llamadas a las funciones de esta se realizan desde la propia aplicación DokuWiki.

b) Diagrama de interfaces



En este diagrama quedan reflejadas las interfaces entre los diferentes componentes y se identifican algunos componentes directamente por su nombre de fichero físico (a diferencia del diagrama anterior). Las únicas interfaces sobre las que interviene nuestro módulo de autenticación son las que aquí se pueden ver (el módulo está representado como ssp.class.php). Estas interfaces son: los métodos SSO de la clase implementada y el API con el Service Provider.

En cuanto a la clase implementada esta se encarga de proporcionar la información de autenticación solicitada por DokuWiki, mediante los método turstExternal y logOff, actualizando las variables necesarias con la información personal del usuario.

En el caso del API con el SP (posible gracias a la carga de la librería de SimpleSAMLphp _autoload.php) se utilizan los métodos requireAuth() y logout() para poder comunicarse con el sistema SAML2 a través del SP.

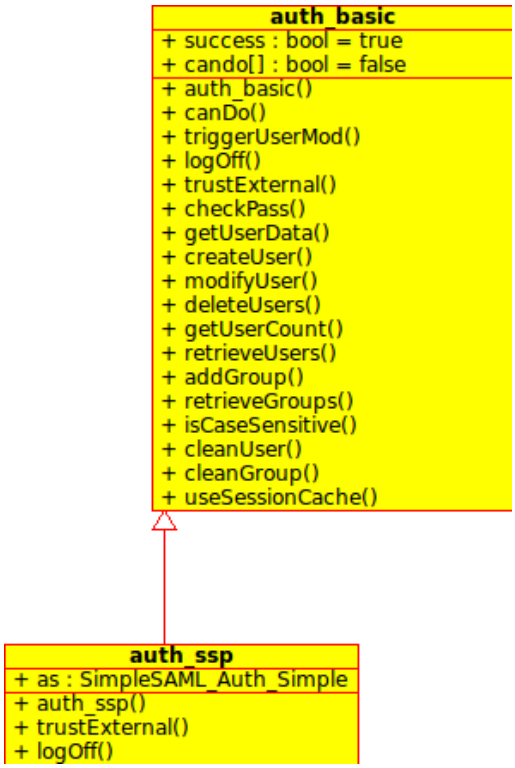
Finalmente la comunicación entre Service Provider e Identity Provider se realiza mediante el protocolo SAML2, y el sistema de almacenamiento de los datos de autenticación se realiza mediante un fichero de configuración de SimpleSAMLphp llamado authsources.php

c) Diagrama de clases

En el diagrama de clases podemos ver como se estructurará nuestro código. La clase a implementar

auth_ssp, deberá heredar de auth_basic puesto que aquí se definen los métodos a implementar en nuestra clase. El constructor de esta clase no hace nada, las variables se inicializan en el momento de su creación. En nuestra clase derivada inicializaremos las variables heredadas $sucess y el array $cando[] en el propio constructor, tal y como se recomienda en los comentarios de la clase base.

Además deberemos crear un atributo $as, que será un objeto de la clase SimpleSAML_Auth_Simple, con el que podremos usar los métodos del API del SP de SimpleSAMLphp.



4.4. Entorno de desarrollo

Se deberá definir:

- entorno tecnológico: hardware, sistema operativo y comunicaciones
- entorno de desarrollo: IDE, compilador, librerías necesarias, generación de documentación de desarrollo
- requisitos de seguridad y otras restricciones de carácter técnico

El entorno tecnológico es un sistema Ubuntu como se ha señalado en fases anteriores. El hardware es un ordenador actual de escritorio de gama media (Intel I3 Core). Las comunicaciones en este caso no son importantes ya que tanto aplicaciones servidor como cliente se comunican a través de la misma máquina. Eso sí, será conveniente una conexión a internet para la consulta de documentación online.

El entorno software para el desarrollo constará de:

- Vim para la implementación del código.
- Un navegador web para pruebas y para consultar documentación: APIs, el manual de PHP, etc.
- El compilador de php y el servidor web Apache.
- Una entidad IdP y una SP correctamente asociadas (mediante metadatos). Las librería de SimpleSAMLphp deberá cargarse en el código PHP para poder comunicarse con estas entidades

mediante API.
- La generación de la documentación de desarrollo se realizará como el resto de la documentación del proyecto, en un formato libre .odt y .pdf que permitirá centralizar toda la documentación en un único documento accesible para todo el mundo.

En cuanto a los requisitos de seguridad, teniendo en cuenta que el desarrollo se produce en un entorno local, cuyo único resultado es la publicación de un módulo (fichero archivador), no se tendrán en cuenta algunas prácticas que sí se deben tener en cuenta en un entorno productivo o accesible desde el exterior. Estos requisitos de seguridad serían:

- La configuración de SimpleSAMLphp en un servidor seguro HTTPS mediante uso de certificados de seguridad.
- La desactivación de cualquier opción de debug que produzca output a través del navegador del usuario.
- En algunos casos la utilización de certificados con ciertos perfiles SAML2 definidos por SimpleSAMLphp.

Como requisito técnico es importante destacar que se activarán en el compilador PHP el modo de aviso mediante Warnings (modo debug), ya que no utilizaremos un IDE que nos avise de errores de sintaxis. Aunque existen plugins de Vim para esto, se considera que con activar la opción de Warnings se obtiene un feedback suficiente. Como se ha dicho antes esta opción nunca puede estar activada en un entorno de producción.

4.5. Especificación de pruebas

Parte del desarrollo será también la definición de las pruebas unitarias que permiten el máximo nivel de independencia entre componentes desarrollados. Como se dijo en el análisis nuestras pruebas se basarán en los casos de uso existentes, y especificarán entradas y salidas como método de verificabilidad (pruebas de caja negra), pero también se tendrá en cuenta la validez del código implementado (pruebas de caja blanca).

El planteamiento de estas pruebas antes del desarrollo del software es importante para que el programador sepa con exactitud que comportamiento se espera de su código. La ejecución de estas pruebas no se llevará a cabo hasta que no se haya desarrollado cada componente sobre los cuales se aplican.

a) Pruebas unitarias

Las pruebas unitarias se pueden automatizar mediante el Test Suite de DokuWiki que utiliza Simple Test Unit Test Framework for PHP[xxiv], sin embargo en nuestro proyecto la implementación de los métodos a testear es casi trivial por lo que se cree más conveniente la especificación de las pruebas unitarias de forma manual e individualizada.

Plan de Pruebas	Método trustExternal() / exitMissingAttribute()
Fecha /Versión / Resp.	/1.0 / J. Hervás
Descripción	Se debe comprobar aisladamente que de obtener atributos vacíos no opcionales (el atributo “grupos” es el único opcional), el método TrustExternal llama a la función exitMissingAttribute. Esta función, como mandan los requerimientos, debe informar al usuario del error de los datos proporionados por el IdP y mostrar un enlace para poder realizar logout, ya que el usuario habrá quedado logueado en IdP.
Estrategia	Pruebas Unitarias. Caja blanca.

Requisitos	Entorno DokuWiki configurado adecuadamente. No se necesita tener el entorno SimpleSAMLphp configurado puesto se trata de pruebas unitarias y no de integración.
Precondiciones	Para la realización de las pruebas se comentarán aquellas líneas del código que hagan referencia a la interfaz del API de SimpleSAMLphp y se fijaran los atributos a recibir por esa parte manualmente

Caso de pruebas	1
Entradas	Se fijan los siguientes atributos con los siguientes valores: $attrs[$ssp_attr['name']][0] = 'nombre_usuario' *$attrs[$ssp_attr['user']][0] = 'id_usuario'*
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"Mail attribute missing from IdP. Please logout to return to login form"*
Caso de pruebas	2
Entradas	Se fijan los siguientes atributos con los siguientes valores: $attrs[$ssp_attr['user']][0] = 'id_usuario' *$attrs[$ssp_attr['mail']][0] = 'mail_usuario'*
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"Name attribute missing from IdP. Please logout to return to login form"*
Caso de pruebas	3
Entradas	Se fijan los siguientes atributos con los siguientes valores: $attrs[$ssp_attr['name']][0] = 'nombre_usuario' *$attrs[$ssp_attr['mail']][0] = 'mail_usuario'*
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"User attribute missing from IdP. Please logout to return to login form"*

Plan de Pruebas	Método trustExternal() / asignación de atributos
Fecha /Versión / Resp.	/1.0 / J. Hervás
Descripción	Se debe comprobar aisladamente que se asignan correctamente los atributos obtenidos a las variables correspondientes de la aplicación DokuWiki y de la sesión PHP y del servidor.
Estrategia	Pruebas Unitarias. Caja blanca.
Requisitos	Entorno DokuWiki configurado adecuadamente. No se necesita tener el entorno SimpleSAMLphp configurado puesto se trata de pruebas unitarias y no de

	integración.
Precondiciones	Para la realización de las pruebas se comentarán aquellas líneas del código que hagan referencia a la interfaz del API de SimpleSAMLphp y se fijaran los atributos a recibir por esa parte manualmente

Caso de pruebas	1
Entradas	Se fijan los siguientes atributos con los siguientes valores: *$attrs[$ssp_attr['name']][0] = 'nombre_usuario'* *$attrs[$ssp_attr['user']][0] = 'id_usuario'* *$attrs[$ssp_attr['mail']][0] = 'mail_usuario'* *$attrs[$ssp_attr['grps']][0] = 'grupo1_usuario'* Además tras las asignaciones, y justo antes de que el método trustExternal retorne se añadirán las siguientes líneas: *echo " USERINFO: ";* *var_dump($USERINFO);* *echo " SERVER['REMOTE_USER']: ";* *var_dump($_SERVER['REMOTE_USER']);* *echo " SESSION: ";* *var_dump($_SESSION);*
Salida esperada	Se deben visualizar los valores de los atributos fijados una vez se loguee el usuario en el formato que proporciona la función var_dump de php

b) Pruebas de integración

Plan de Pruebas	Método TrustExternal
Fecha /Versión / Resp.	/1.0 / J. Hervás
Descripción	Comprobar que los permisos ACL funcionan correctamente, es decir el nombre de usuario y los grupos a los que pertenece este son recuperados correctamente, y por tanto la configración ACL puede funcionar en base a estos. Se debe incluir un caso de prueba al menos con caracteres extraños en el grupo y el id. de usuario. Comprobar también que los atributos vacíos obtenidos de IdP, de obtención obligatoria, provocan error. Por otro lado, en cualquier otro caso el procedimiento siempre debe terminar devolviendo true. NOTA: Los parámetros del método siempre vienen vacíos (por lo que no se comprueban), los atributos se obtienen mediante RequireAuth() al tratase de un método de autenticación externo.
Estrategia	Pruebas de integración. Caja negra. Se definen unos casos límite a probar.
Requisitos	Entorno SimpleSAMLphp y DokuWiki configurado adecuadamente.

Precondiciones	Deben existir usuarios dados de alta en el Proveedor de Identidad, uno de ellos debe corresponder al usuario administrador de DokuWiki y deben tener definidos los atributos que se requieran para realizar las pruebas especificadas para cada caso a continuación. Todos los usuarios pertenecen al grupo 'user' por defecto con el plugin de autenticación 'plain', para imitar este comportamiento en el entorno SSO, se autogenera el grupo 'user' mediante la siguiente línea en *config.php*: *// All users will be members of 'users' and 'members'* *362 61 => array('class' => 'core:AttributeAdd', 'eduPersonAffiliation' => array('user')),* Se ha configurado el fichero acl.auth.php de DokuWiki de la siguiente forma: ** @ALL 0 # ningun permiso* ** @user 1 # lectura* ** person1 8 # lectura, edición, creación y upload* ** @student 2 # lectura y edición* ** @edición 16 # todos los permisos: lectura, edición, creación, upload, eliminación de ficheros subidos* De forma que se dan permisos generales para todas las ubicaciones del Wiki (“*”), haciendo diferencias por grupos y por el usuario “person1”. El número corresponde a los permisos que indica el comentario

Caso de pruebas	1
Entradas	Se define el usuario siguiente en *authsources.php*: *'admin:<contraseña>' => array(* *'uid' => array('admin'),* *'cn' => array('DokuWiki Administrator'),* *'email' => array('webmaster@localhost'),* *'eduPersonAffiliation' => array('admin'),* *// usuario administrador (pertenece al grupo admin)* *),* luego se introduce “admin” y <contraseña> en el formulario de entrada de datos y se hace login
Salida esperada	Una vez autenticado el usuario debe poder acceder a las opciones de autenticación mediante un enlace “Admin” situado en la pantalla principal
Caso de pruebas	2
Entradas	Se define el usuario siguiente en *authsources.php*: *'person1:personpass' => array(* *'uid' => array('person1'),* *'cn' => array('person 1'),* *'email' => array('person1@gm.com'),* *'eduPersonAffiliation' => array('student'),* *// usuario que pertenece al grupo student* *),* luego se introduce “person1” y “personpass” en el formulario de entrada de datos y se hace login
Salida esperada	Una vez autenticado el usuario debe tener permisos de *lectura, edición, creación y upload* en todo el wiki

Caso de pruebas	3
Entradas	Se define el usuario siguiente en *authsources.php*: *'person2:personpass' => array(* *'uid' => array('person2'),* *'cn' => array('person 2'),* *'email' => array('person2@gm.com'),* *// no incluimos el atributo eduPersonAffiliation* *),* luego se introduce “person2” y “personpass” en el formulario de entrada de datos y se hace login
Salida esperada	Una vez autenticado el usuario debe tener permisos de *lectura* (por pertenecer al grupo user por defecto)
Caso de pruebas	4
Entradas	Se define el usuario siguiente en *authsources.php*: *'person3:personpass' => array(* *'uid' => array('person3'),* *'cn' => array('person 3'),* *'eduPersonAffiliation' => array('student'),* *// no incluimos atributo email* *),* luego se introduce “person3” y “personpass” en el formulario de entrada de datos y se hace login
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *“Mail attribute missing from IdP. Please logout to return to login form”*
Caso de pruebas	5
Entradas	Se define el usuario siguiente en *authsources.php*: *'person4:personpass' => array(* *'uid' => array('person4'),* *'email' => array('person4@gm.com'),* *'eduPersonAffiliation' => array('student'),* *// no incluimos atributo cn* *),* luego se introduce “person4” y “personpass” en el formulario de entrada de datos y se hace login
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *“Name attribute missing from IdP. Please logout to return to login form”*
Caso de pruebas	6
Entradas	Se define el usuario siguiente en *authsources.php*: *'person5:personpass' => array(* *'cn' => array('person 5'),* *'email' => array('person5@gm.com'),* *'eduPersonAffiliation' => array('student'),* *// no incluimos atributo uid* *),*

	luego se introduce “person5” y “personpass” en el formulario de entrada de datos y se hace login
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *“User attribute missing from IdP. Please logout to return to login form”*
Caso de pruebas	7
Entradas	Se define el usuario siguiente en *authsources.php*: *'person6àéïôúñ:personpass' => array(* *'uid' => array('person6àéïôúñ'),* *'cn' => array('person 6 àéïôúñ'),* *'email' => array('person6@gm.com'),* *'eduPersonAffiliation' => array('student','edición'),* *// contiene carácteres "extraños"* *),* luego se introduce “person6” y “personpass” en el formulario de entrada de datos y se hace login
Salida esperada	Una vez autenticado el usuario debe tener permisos de *lectura, edición, creación, upload, delete* en todo el wiki

Plan de Pruebas	Método trustExternal / autenticación previa en IdP
Fecha /Versión / Resp.	/1.0 / J. Hervás
Descripción	Comprobar que el usuario que ha hecho login en IdP accede a DokuWiki sin la necesidad de introducir datos de login
Estrategia	Pruebas de integración. Caja negra.
Requisitos	Entorno SimpleSAMLphp y DokuWiki configurado adecuadamente.
Precondiciones	Debe existir al menos un usuario dado de alta en el Proveedor de Identidad. El usuario se debe autenticar en IdP previamente, pero no a través de DokuWiki, por ejemplo mediante la interfaz que provee simpleSAMLphp (http://localhost/simplesaml/*)

Caso de pruebas	8
Entradas	Acceder a cualquier URL de DokuWiki
Salida esperada	El usuario no recibe el formulario de login de DokuWiki, sino directamente accede a la sesión con los atributos proporcionados por el IdP

Plan de Pruebas	Método LogOff
Fecha /Versión /	/1.0 / J. Hervás

Resp.	
Descripción	Comprobar que el usuario que ha hecho login en IdP puede cerrar su sesión en IdP a través de DokuWiki
Estrategia	Pruebas de integración. Caja negra.
Requisitos	Entorno SimpleSAMLphp y DokuWiki configurado adecuadamente.
Precondiciones	Debe existir al menos un usuario dado de alta en el Proveedor de Identidad.

Caso de pruebas	9
Entradas	Hacer click sobre el enlace de logout en DokuWiki
Salida esperada	El usuario cierra la sesión tanto en DokuWiki como en el IdP

4.6. Requisitos de implantación. Publicación del código

La implantación del módulo queda fuera del alcance del proyecto. En lugar de esto el código se publicará bajo una licencia GPL V2. Para ello se deberá tener en cuenta la inclusión del siguiente anuncio de la licencia en un fichero en la raíz del paquete junto con el código:

SSP. Módulo de autenticación SAML2 para DokuWiki mediante SimpleSAMLphp
Copyright (C) 2011 Jorge Hervás Vigueras

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

5. Desarrollo

El desarrollo del código se ha llevado a cabo en la medida de lo posible en paralelo junto con la fase anterior. De esta forma una vez definidos los objetivos de diseño y pruebas para la clase php se ha implementado dejando para más tarde el diseño de otros componentes.

A continuación se expondrá detalladamente como se ha realizado este desarrollo, que se ha compuesto de la planificación del mismo, su ejecución, y la redacción de la documentación de usuario.

5.1. Planificación del desarrollo e integración

Según el convenio de colaboración educativa a firmar con la empresa PriSE la planificación del trabajo de prácticas externas se limita a 185 horas.

Hay que tener en cuenta sin embargo que algunas de las tareas aquí previstas como la entrega de documentación de las fases de estudio de viabilidad, análisis y diseño se realizará en cierta medida en la asignatura paralela de TFM.

Es por ese motivo que en el diagrama de Gantt se han minimizado las horas empleadas en estas fases en relación con las tareas posteriores como el desarrollo que si se debe realizar de forma exclusiva para esta asignatura.

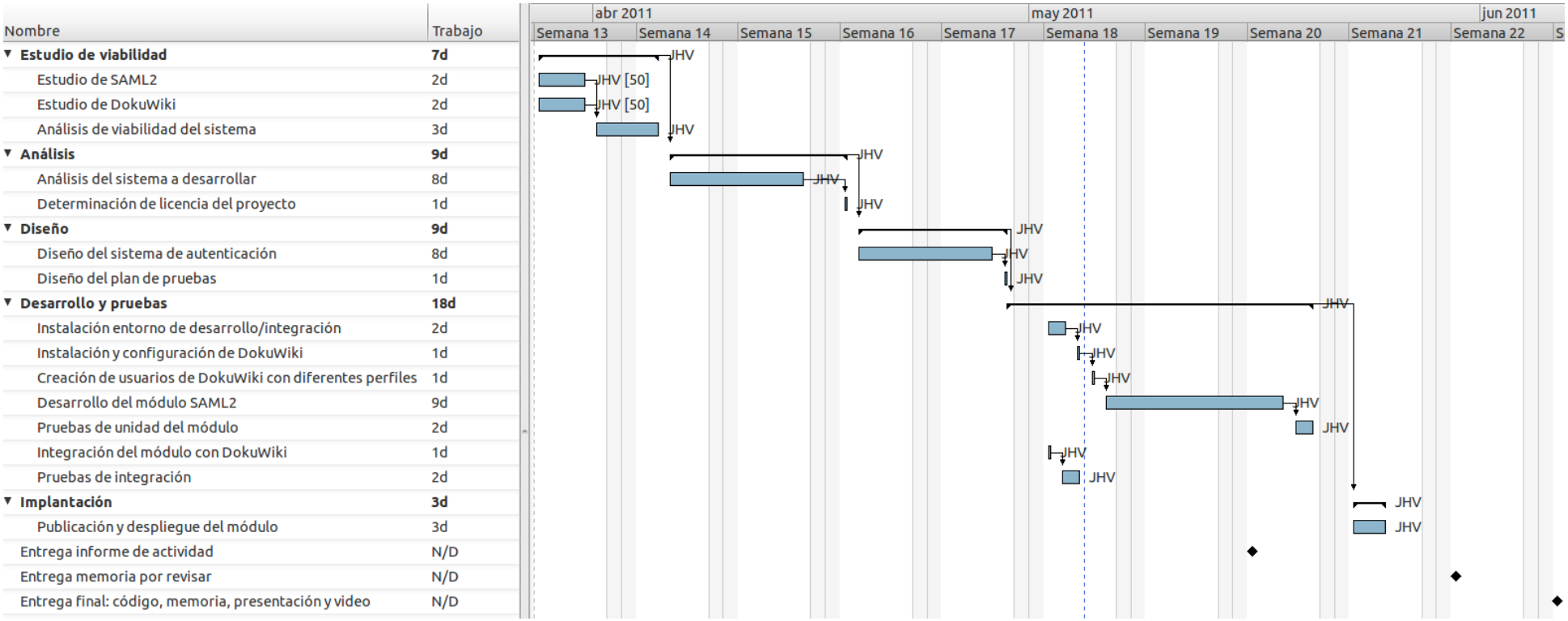


Diagrama de Gantt del proyecto final de máster

En el diagrama anterior se pueden ver la duración relativa de las tareas representadas mediante las clásicas barras de Gantt que indican en que fechas se desarrollan estas.

Para ello se tiene en cuenta que el calendario marcado no contempla trabajo en festivos, así como se ha fijado una jornada laboral de 4 horas, resultado de distribuir las 185 horas asignadas sobre el calendario hasta principios de junio (donde termina el semestre aproximadamente).

Como solamente se cuenta con un desarrollador las tareas se realizan de forma secuencial a lo largo de las fases (a excepción de las 2 tareas de estudio inicial que se pueden realizar en paralelo al 50%)

También podemos ver representados con un cuadrado de lado los hitos (entregas): entrega de un informe de actividad, y la entrega final que incluye toda la documentación (ver metodología) y código del proyecto.

5.2. Preparación del entorno de desarrollo. Instalación de los componentes principales

Tal y como se vio en el inicio del estudio de viabilidad, apartado “Versiones de los componentes de terceros”, se utilizarán los siguientes componentes de terceros en la preparación del entorno de desarrollo:

– DokuWiki, versión 0.0.20091225c-8, bajada de los repositorios de Ubuntu 10.10

– SimpleSAMLphp, versión estable 1.8

– Php, versión 5.3.3, con Apache, versión 2.2.16-1ubuntu3.1, bajado de los repositorios de Ubuntu 10.10

La descripción de la instalación de los componentes hace referencia a un sistema Ubuntu 10.10. (Aunque en otros sistemas Debian/Ubuntu podría ser válida también).

a) Instalación de DokuWiki

La instalación de DokuWiki se realiza mediante la instalación del paquete “dokuwiki”, instalable en línea de comandos mediante:

sudo apt-get install dokuwiki

Para saber donde se han instalado los archivos del paquete instalado se pueden ejecutar comandos como “dpkg -s” que nos dice donde se encuentran los archivos de configuración, o “dpkg –contents” que nos muestra el contenido del paquete (todas las rutas donde se instalan paquetes).

Habiendo ejecutado estos comandos vemos que los ficheros se han instalado en 3 rutas distintas conforme a la “metodología” del sistema de directorios de Debian, que son:

– /etc/dokuwiki

– /var/lib/dokuwiki/

– /usr/share/dokuwiki/

En /etc/dokuwiki encontramos ficheros de configuración, en /var/lib/dokuwiki/data encontraremos los datos, y en /usr/share/dokuwiki la raíz del servidor web. Por tanto al seguir la documentación de DokuWiki deberemos tener en cuenta que cuando se refiere al directorio “config” en nuestro caso se trata de “/etc/dokuwiki”, etc.

Por último, a causa de un bug, hemos cambiado una línea en el fichero /etc/apache2/conf.d/dokuwiki.conf, tal y como se explica en las siguientes instrucciones específicas para el paquete de Ubuntu: http://www.dokuwiki.org/install:ubuntu

b) Instalación de SimpleSAMLphp

Para la instalación de SimpleSAMLphp se han seguido las instrucciones de instalación genéricas documentadas en el proyecto[xxv]. El motivo de no utilizar la instalación de los repositorios, fue que la versión

en ellos era bastante antigua y además de algunos problemas en ella no se proporcionaba tanta documentación como para la versión actualmente estable.

La versión actual 1.8, estable se ha bajado de aquí: http://code.google.com/p/simplesamlphp/downloads/list

Una vez descomprimido el fichero .tar.gz y situado en /var/simplesamlphp, se ha añadido la siguiente línea en el fichero de configuración de virtual hosts, en nuestro caso en */etc/apache2/sites-enabled/000-default*:

Alias /simplesaml /var/simplesamlphp/www

Finalmente se debe configurar inicialmente la contraseña administrativa, así como algunos datos de administrador en el fichero config/config.php de simpleSAMLphp:

'auth.adminpassword' => '<admin_pass>',

'secretsalt' => '<secret_salt>',

'technicalcontact_name' => 'J. H.',

'technicalcontact_email' => 'jordihv@gmail.com',

'timezone' => 'Europe/Madrid',

c) Instalación de Apache y Php

Para su instalación hemos instalado un único meta-paquete que se encargara de instalar todas las dependencias requeridas por él mismo, incluyendo el servidor Apache. El paquete a instalar es “php5”, que se puede instalar desde línea de comandos mediante la siguiente instrucción:

sudo apt-get install php5

5.3. Preparación del entorno de desarrollo. Configuración de los componentes principales

Una vez instalados los componentes principales DokuWiki y SimpleSAMLphp, será necesaria tanto la configuración de estos para su funcionamiento por separado, como la configuración que permita la comunicación entre ambos componentes.

a) Configuración SimpleSAMLphp

La configuración de SimpleSAMLphp nos permitirá activar una instancia SP y una instancia IdP sobre la misma máquina. Así mismo estas instancias deberán poder comunicarse entre ellas para lo que necesitarán compartir unos metadatos que permitan la confianza mutua entre ellas.

Para configurar la instancia SP, hemos debido añadir las siguientes líneas en *config/authsources.php*:

```
'default-sp' => array(
    'saml:SP',
```

de esta manera hemos asignado el protocolo SAML2 a nuestro Service Provider (no hay que olvidar que SimpleSAMLphp también permite el protocolo Shibboleth).

El siguiente paso consiste en relacionar a nuestro Service Provider con la instancia IdP que configuraremos posteriormente. Para ello deberemos incluir sus datos identificativos (localización) y metadatos en `metadata/saml20-idp-remote.php`. En este caso son:

```
$metadata['http://localhost/simplesaml/saml2/idp/metadata.php'] = array (
  'metadata-set' => 'saml20-idp-remote',
  'entityid' => 'http://localhost/simplesaml/saml2/idp/metadata.php',
  'SingleSignOnService' => 'http://localhost/simplesaml/saml2/idp/SSOService.php',
  'SingleLogoutService' => 'http://localhost/simplesaml/saml2/idp/SingleLogoutService.php',
                                                    'certData'                    =>
```

'MIICgTCCAeoCCQCbOlrWDdX7FTANBgkqhkiG9w0BAQUFADCBhDELMAkGA1UEBhMCTk8xGDAWBgN
VBAgTD0FuZHJlYXMgU29sYmVyZzEMMAoGA1UEBxMDRm9vMRAwDgYDVQQKEwdVTklORVRUMRgw
FgYDVQQDEw9mZWlkZS5lcmxhbmcubm8xITAfBgkqhkiG9w0BCQEWEmFuZHJlYXNAdW5pbmV0dC5ubz
AeFw0wNzA2MTUxMjAxMzVaFw0wNzA4MTQxMjAxMzVaMIGEMQswCQYDVQQGEwJOTzEYMBYGA1U
ECBMPQW5kcmVhcyBTb2xiZXJnMQwwCgYDVQQHEwNGb28xEDAOBgNVBAoTB1VOSU5FVFQxGDAW
BgNVBAMTD2ZlaWRlLmVybGFuZy5ubzEhMB8GCSqGSIb3DQEJARYSYW5kcmVhc0B1bmluZXR0Lm5vM
IGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDivbhR7P516x/S3BqKxupQe0LONoliupiBOesCO3SH
bDrl3+q9IbfnfmE04rNuMcPsIxB161TdDpIesLCn7c8aPHISKOtPlAeTZSnb8QAu7aRjZq3+PbrP5uW3TcfCGP
tKTytHOge/OlJbo078dVhXQ14d1EDwXJW1rRXuUt4C8QIDAQABMA0GCSqGSIb3DQEBBQUAA4GBACD
Vfp86HObqY+e8BUoWQ9+VMQx1ASDohBjwOsg2WykUqRXF+dLfcUH9dWR63CtZIKFDbStNomPnQz7nb
K+onygwBspVEbnHuUihZq3ZUdmumQqCw4Uvs/1Uvq3orOo/WJVhTyvLgFVK2QarQ4/67OZfHd7R+POBXh
ophSMv1ZOo',

'NameIDFormat' => 'urn:oasis:names:tc:SAML:2.0:nameid-format:transient',

);

Esta información ha sido extraída de http://localhost/simplesaml/saml2/idp/metadata.php?output=xhtml una vez se ha configurado IdP.

También como un atributo más del array 'default-sp' se ha añadido la línea siguiente que permite asignar un IdP por defecto en el caso de tener más de uno:

'idp' => 'http://localhost/simplesaml/saml2/idp/metadata.php'

Para la activación del Identity Provider se ha puesto a *true* el siguiente atributo de la configuración en config/config.php:

'enable.saml20-idp' => true,

De la misma forma que cada SP necesita identificar su/s IdP debemos identificar a que SP vamos a conectar el servicio del IdP que se configuran. Para esto hemos modificado el fichero *metadata/saml20-sp-remote.php* añadiendo las líneas:

$metadata['http://localhost'] = array (

'AssertionConsumerService' => 'http://localhost/simplesaml/module.php/saml/sp/saml2-acs.php/default-sp',

'SingleLogoutService' => 'http://localhost/simplesaml/module.php/saml/sp/saml2-logout.php/default-sp',

);

Estas líneas han sido extraídas de *http://localhost/simplesaml/module.php/saml/sp/metadata.php/default-sp?output=xhtml.*

Llegados a este punto se han conectado los datos de autenticación al IdP, para ello basta con incluir un array con los usuarios y atributos correspondientes que se necesitan. Esto se ha hecho de manera estática, puesto que únicamente se necesitan para el desarrollo de las pruebas posteriores.

A continuación se muestran unas líneas que definen a varios usuarios en el fichero */var/simplesamlphp/config/authsources.php*:

```
'example-userpass' => array(
      'exampleauth:UserPass',
      'admin:XXXX' => array(
        'uid' => array('admin'),
        'cn' => array('DokuWiki Administrator'),
```

```
        'email' => array('webmaster@localhost'),
        'eduPersonAffiliation' => array('admin','user'),
      ),
      'person1:personpass' => array(
        'uid' => array('person1'),
        'cn' => array('employee 1'),
        'email' => array('examplemail3@gm.com'),
      ),
[...]
```

En este ejemplo podemos ver como quedan definidos el nombre de usuario y contraseña a introducir en el formulario de login (admin, es el usuario y XXXX la contraseña), y los atributos del usuario con sus correspondientes valores.

Hay que señalar que el nombre de los atributos deberá ser exactamante el que se puede ver aquí: "uid", "cn", "email", "eduPersonAffiliation" (a.k.a grupo en DokuWiki). De no respetar el nombre de estos atributos, por algún motivo (por ejemplo, por que los datos de autenticación que se sirvan también sirvan a otros SP) deberá cambiarse la configuración del módulo de autenticación (en el fichero ssp.class.php), que por defecto incluye las siguientes líneas:

```
// configure attribute names to match the ones used by our authentication backend (IdP)
$ssp_attr['name'] = 'cn';
$ssp_attr['user'] = 'uid';
$ssp_attr['mail'] = 'email';
$ssp_attr['grps'] = 'eduPersonAffiliation';
```

Para activar este sistema de información de autenticación, deberemos ejecutar en una línea de comandos en la raíz de la instalación de SimpleSAMLphp:

touch modules/exampleauth/enable

además de asignar al atributo de configuración auth, el método mencionado en *metadata/saml20-idp-hosted.php:*

```
'auth' => 'example-userpass',
```

b) Configuración de DokuWiki

DokuWiki permite el uso de diferentes backends de autenticación, y la definición de nuevos backends definiendo una nueva clase guardada en un fichero propio con un nombre del tipo <backend>.class.php

Para utilizar un nuevo método de autenticación que implementemos es necesario especificarlo en la configuración de DokuWiki. Para ello debemos cambiar la opción actual *plain* por el nombre del backend que hayamos especificado en el fichero que contiene la clase en las opciones de DokuWiki (fichero /etc/dokuwiki/dokuwiki.php):

```
#$conf['authtype']    = 'plain';        //which authentication backend should be used
$conf['authtype']    = 'ssp';          //which authentication backend should be used
```

c) Configuración para la integración de DokuWiki y simpleSAMLphp

Con este cambio realizado sin embargo, aún no es posible la comunicación entre DokuWiki y simpleSAMLphp debido al uso de una cookie de sesión php por parte de ambos. Esto se puede arreglar de varias maneras, la más sencilla de las cuales es el uso de *memcache* para el registro de la información del usuario en lugar de la sesión php que se usa por defecto en simpleSAMLphp. Para el uso de esta solución se requiere tener instalados los paquetes *memcached* y *php5-memcache*, y modificar la siguiente línea en

/var/simplesamlphp/config/config.php:

#'store.type' => 'phpsession',
'store.type' => 'memcache',

La otra solución implica comentar 2 líneas en el fichero init.php de DokuWiki para que no se establezcan los parametros de la Cookie que hacen imposible compartirla con simpleSAMLphp, de igual forma hay que cambiar el nombre de la cookie en *config/config.php* de simpleSAMLphp para que quede establecida con el mismo nombre que le da DokuWiki y pueda ser compartida:

'session.phpsession.cookiename' => 'DokuWiki',

5.4. Desarrollo del código

a) Formato de la clase backend de autenticacion

El formato del nombre del fichero donde se implementa la clase correspondiente al nuevo backend de autorización de SimpleSAMLphp debe seguir el formato indicado en la documentación[xxvi]:

<backend>.class.php

Así mismo, dentro de ese fichero debe existir una clase que extienda auth_basic u otro backend de autenticación cuyo nombre debe ser del tipo:

auth_<backend>

La implementación de los diferentes métodos existentes en auth_basic (clase base que utilizaremos nosotros) dependerá de que características pueda realizar nuestro backend. Esto se establecerá en el propio constructor de la clase determinando que otros métodos se deben implementar o no.

b) Implementación de la clase derivada

El servicio SSO proporcionado por SimpleSAMLphp tiene la característica de ser una capa de abstracción para la autenticación con respecto al sistema DokuWiki. Esto implica que DokuWiki sólo tiene acceso a los datos de autenticación para dos procedimientos: la autenticación, y el cierre de sesión (logoff).

A continuación se presentan los métodos implementados con comentarios en inglés (según decisión de diseño), que contiene la definición de los métodos trustExternal() y logoff() que se corresponden con los procedimientos señalados:

```
function auth_ssp() { //constructor
  // we set the features of our authentication backend to TRUE, the base class defaults to FALSE the rest
  $this->cando['external'] = true;
  $this->cando['logoff']   = true;
  $this->success = true;
}

function trustExternal($user,$pass,$sticky=false){
  global $USERINFO;
  global $conf;

  $sticky ? $sticky = true : $sticky = false; //sanity check

  // loading of simplesamlphp library
  require_once($conf['ssp_path'] . '/lib/_autoload.php');

  // create auth object and use api to require authentication and get attributes
```

```
    $this->as = new SimpleSAML_Auth_Simple('default-sp');

     // the next line should be discommented to enable guest users (not authenticated) enter DokuWiki, see
also documentation
    # if ($this->as->isAuthenticated()) {

    $this->as->requireAuth();
    $attrs = $this->as->getAttributes();

    // check for valid attributes (not empty) and update USERINFO var from dokuwiki
    if (!isset($attrs[$conf['ssp_attr_name']][0])) {
      $this->exitMissingAttribute('Name');
    }
    $USERINFO['name'] = $attrs[$conf['ssp_attr_name']][0];

    if (!isset($attrs[$conf['ssp_attr_mail']][0])) {
      $this->exitMissingAttribute('Mail');
    }
    $USERINFO['mail'] = $attrs[$conf['ssp_attr_mail']][0];

    // groups may be empty (by default any user belongs to the user group) don't perform empty check
    $USERINFO['grps'] = $attrs[$conf['ssp_attr_grps']];

    if (!isset($attrs[$conf['ssp_attr_user']][0])) {
      $this->exitMissingAttribute('User');
    }

    // asign user id to the user global information
    $_SERVER['REMOTE_USER'] = $attrs[$conf['ssp_attr_user']][0];

    // assign user id and the data from USERINFO to the DokuWiki session cookie
    $_SESSION[DOKU_COOKIE]['auth']['user'] = $attrs[$conf['ssp_attr_user']][0];
    $_SESSION[DOKU_COOKIE]['auth']['info'] = $USERINFO;

    # } // end if_isAuthenticated()

    return true;
  }

  function exitMissingAttribute( $attribute ){
    // get logout link
    $url = $this->as->getLogoutURL();
    $logoutlink = '<a href="' . htmlspecialchars($url) . '">logout</a>';
    die( $attribute . ' attribute missing from IdP. Please ' . $logoutlink . ' to return to login form');
  }

  function logOff(){
    // use the simpleSAMLphp authentication object created in trustExternal to logout
    $this->as->logout('/');
  }

}
```

Para la interacción entre el backend de autenticación (funciones implementadas) y las instancias IdP y SP mediante el API de SimpleSAMLphp es imprescindible la carga de la librería de SimpleSAMLphp en primer lugar, esta se encuentra en: <ruta de simplesamlphp>/lib/_autoload.php. Debido a que esta ruta puede variar según la instalación está ruta será configurable (ver apartado de “Documentación de usuario”.

La implementación del código ha supuesto además de la lectura de la documentación correspondiente a DokuWiki (para saber donde almacenar la información de autenticación), un repaso sobre la sintaxis del lenguaje php, y temas relacionados como es la utilización de cookies y variables de sesión. También ha sido clave la lectura de los comentarios en la clase base, lo que ha facilitado la comprensión de las funciones a implementar en la clase derivada.

Las variables globales de asignación obligatoria (como se puede ver en el código) son las siguientes[xxvii]:

$USERINFO['name']
$USERINFO['mail']
$USERINFO['grps']
$_SERVER['REMOTE_USER']
$_SESSION[DOKU_COOKIE]['auth']['user']
$_SESSION[DOKU_COOKIE]['auth']['info']

Esto responde a la necesidad de:

- Guardar datos de usuario en variable global $USERINFO
- Guardar usuario en $_SERVER['REMOTE_USER']
- Guardar usuario y datos del mismo ($USERINFO) en $_SESSION[DOKU_COOKIE]

La asignación de la variable $USERINFO['grps'] puede ser un array vacío.

5.5. Ejecución de pruebas de unidad y pruebas de integración

A continuación se muestran los resultados de la ejecución de las pruebas desarrolladas en la fase de diseño (pruebas unitarias e integradas).

a) Pruebas unitarias

Plan de Pruebas	Método trustExternal() / exitMissingAttribute()	
Caso de pruebas	1	**Resultado obtenido**
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"Mail attribute missing from IdP. Please logout to return to login form"*	OK. Se ha obtenido la salida esperada
Caso de pruebas	2	
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"Name attribute missing from IdP. Please logout to return to login form"*	OK. Se ha obtenido la salida esperada
Caso de pruebas	3	
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"User attribute missing from IdP. Please logout to return to login form"*	OK. Se ha obtenido la salida esperada

Plan de Pruebas	Método trustExternal() / asignación de atributos
Caso de pruebas	1
Salida esperada	Se deben visualizar los valores de los atributos fijados una vez se loguee el usuario en el formato que proporciona la función var_dump de php

b) Pruebas de integración

Plan de Pruebas	Método TrustExternal	
Caso de pruebas	1	**Resultado obtenido**
Salida esperada	Una vez autenticado el usuario debe poder acceder a las opciones de autenticación mediante un enlace "Admin" situado en la pantalla principal	OK. Se ha obtenido la salida esperada
Caso de pruebas	2	
Salida esperada	Una vez autenticado el usuario debe tener permisos de *lectura, edición, creación y upload* en todo el wiki	OK. Se ha obtenido la salida esperada
Caso de pruebas	3	
Salida esperada	Una vez autenticado el usuario debe tener permisos de *lectura* (por pertenecer al grupo user por defecto)	OK. Se ha obtenido la salida esperada
Caso de pruebas	4	
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"Mail attribute missing from IdP. Please logout to return to login form"*	OK. Se ha obtenido la salida esperada
Caso de pruebas	5	
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"Name attribute missing from IdP. Please logout to return to login form"*	OK. Se ha obtenido la salida esperada
Caso de pruebas	6	
Salida esperada	Se debe obtener un mensaje en pantalla que diga (y el enlace debe permitir hacer logout): *"User attribute missing from IdP. Please logout to return to login form"*	OK. Se ha obtenido la salida esperada
Caso de pruebas	7	
Salida esperada	Una vez autenticado el usuario debe tener permisos de *lectura, edición, creación, upload, delete* en todo el wiki	OK. Se ha obtenido la salida esperada

Plan de Pruebas	Método trustExternal / autenticación previa en IdP	
Caso de pruebas	8	**Resultado obtenido**
Salida esperada	El usuario no recibe el formulario de login de DokuWiki, sino directamente accede a la sesión con los atributos	OK. Se ha obtenido la salida esperada

	proporcionados por el IdP	

Plan de Pruebas	Método LogOff	
Caso de pruebas	9	**Resultado obtenido**
Salida esperada	El usuario cierra la sesión tanto en DokuWiki como en el IdP	OK. Se ha obtenido la salida esperada

6. Publicación del código y mantenimiento

6.1. Publicación del código en dokuwiki.org

La publicación del código se ha realizado dentro del wiki de DokuWiki de forma que los usuarios de este sistema puedan acceder fácilmente al presente desarrollo.

En el wiki oficial de DokuWiki se pueden crear libremente nuevos plugins creando nuevas páginas en el namespace de plugins, como se puede ver en:
http://www.dokuwiki.org/devel:plugins#publishing_a_plugin_on_dokuwikiorg.

Sin embargo como nuestro módulo no se ha podido adaptar al formato de plugins se ha debido buscar otro lugar donde poder publicarlo.

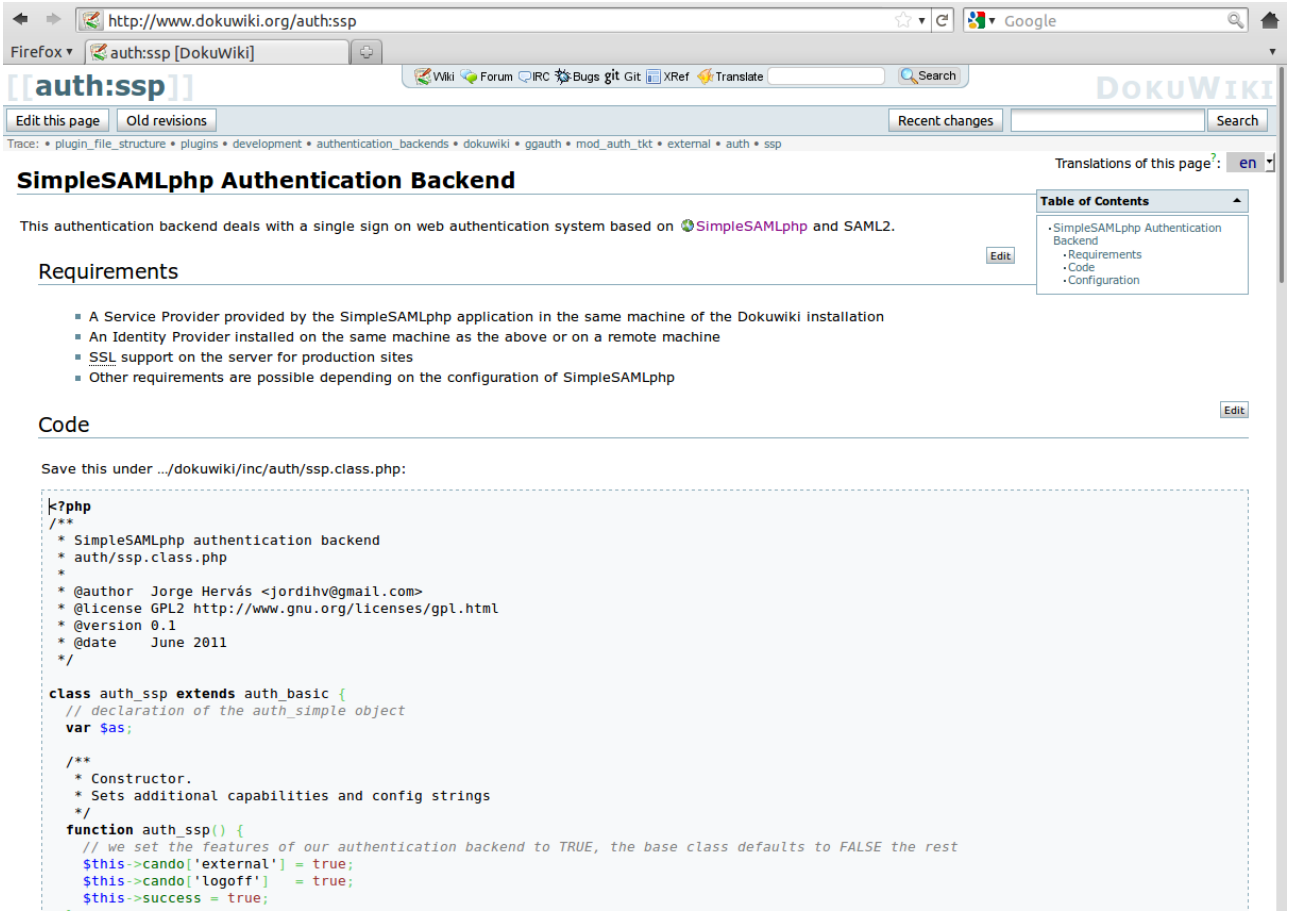
Para ello se ha encontrado la página del mismo wiki:
http://www.dokuwiki.org/auth#contributed_backends , donde encontramos backends de autenticación como el nuestro, con instrucciones de instalación y el propio código anidado en la página.

Para indicar la licencia y la autoría incluimos un encabezamiento en el código similar a uno de los que ahí podemos encontrar. Por ejemplo:

```
/**
 * PAM authentication backend
 * @author  Michael Gorven <michael003+dokuwiki@gmail.com>
 * @license GPL2 http://www.gnu.org/licenses/gpl.html
 * @version 0.2
 * @date    March 2008
 */
```

De la misma forma se incluyen las instrucciones de instalación y de configuración mediante el uso adecuado de la sintaxis del wiki (sobretodo separando claramente las partes que son código literal de las explicaciones).

La url del wiki es http://www.dokuwiki.org/auth:ssp.



Captura de pantalla de la página del wiki con el código y las instrucciones de instalacion

El código del proyecto se incluirá de todas formas junto a la memoria en el formato previsto por la normativa de presentación de documentación del Proyecto Final de Máster.

6.2. Mantenimiento

Como consecuencia de la naturaleza abierta de wiki de DokuWiki, el código queda disponible para la modificación posterior por parte de los usuarios del sitio, de forma que cualquiera pueda acceder al mantenimiento del mismo.

Como método para asegurar que el código es versionado correctamente tras cada edición (cosa que no podría verse directamente a través de DokuWiki), se puede utilizar un repositorio online tipo https://gist.github.com/ para la publicación y compartición de fragmentos de código.

7. Conclusiones

Se han conseguido los objetivos principales previstos en los requerimientos, es decir aquellos que hacían referencia a la funcionalidad de nuestro módulo de autenticación: se puede hacer login y logout del proveedor de Identidad, ya sea desde DokuWiki o no, y el sistema DokuWiki se encuentra perfectamente integrado con este sistema.

Jorge Hervás Vigueras

Por otra parte no se ha podido implementar el código en el formato de plugin que se había previsto inicialmente porque se ha considerado que el sistema introducía más inconvenientes que ventajas respecto a una instalación manual. Más que un objetivo no cumplido este hecho se debe considerar una decisión de diseño.

La toma de decisiones a lo largo del proyecto ha tratado de seguir el orden establecido en un modelo de desarrollo consistente en el ciclo clásico de desarrollo de proyectos. Esto ha permitido el refinamiento de la solución de forma progresiva a lo largo de las distintas fases.

No obstante, se han detectado algunos errores en la definición de fases anteriores que se han tenido que ir corrigiendo a posteriori, conformando un modelo iterativo en lo que concierne a algunos puntos aislados. Concretamente estos han sido:

- La no incorporación en el análisis del requerimiento de proveer un enlace de logout en caso de error en el procedimiento de autenticación (ya que el usuario queda logueado en IdP igualmente). Este es un error de falta de detalle en los requerimientos.

- La eliminación del requerimiento de la implementación de un plugin conforme a las especificaciones del proyecto DokuWiki. El planteamiento de la inclusión de la implementación de un plugin fue al principio solamente una posibilidad, pero se terminó por incluir en los requerimientos sin estudiar convenientemente las ventajas e inconvenientes que podía aportar.

- La redacción incorrecta de algunos detalles en los casos de uso, concretamente en lo que concierne a la forma en que se procede al login por falta de documentación específica.

Estos errores no han supuesto más retraso que el de la revisión de detalles en la documentación de fases anteriores y por tanto no se pueden considerar como errores graves. Sin embargo, al analizarlos todos parecen tener el mismo denominador común: la falta de alguna información sensible en el momento de la realización del análisis del sistema.

Como conclusión para evitar este tipo de problemas en un futuro sería precisa la realización de pruebas mediante la instalación previa de los componentes, durante la fases de análisis o antes, para obtener esa información que pueda faltar, mediante la propia experiencia.

En ocasiones pueden existir ciertas lagunas en la documentación de un componente o en la integración de varios, ya sea por errores en la documentación o por la naturaleza de integración de componentes diversos del software libre. Sin embargo, hay que destacar que la naturaleza del software libre nos permite también, más allá de la existencia o no de documentación en un punto concreto, poder entender ciertos procesos mediante la simple lectura del código y sus comentarios.

En cuanto a posibles ampliaciones futuras: El módulo de autenticación SAML2 presentado basa su sistema de autenticación en una fuente de autenticación externa a DokuWiki, otorgando completamente el control de la autenticación al Proveedor de Identidad SAML2. Esto provoca, colateralmente, que no se permita el acceso de ningún usuario que no haya introducido sus credenciales (cuyo control de acceso se controla mediante ACL y el grupo @ALL), por contra de lo que permite el comportamiento típico de DokuWiki con el bakend de autenticación original.

Esta característica, conocida como “Lazy Authentication”, está fuera de los límites de este proyecto ya que nuestro módulo tiene la funcionalidad exclusiva de proporcionar una autenticación mediante SAML2, y no un sistema en cierta medida híbrido como el descrito. Sin embargo sí que se ha dejado en el código del módulo un comentario referente a esta posibilidad de forma que se pueda tener en cuenta en un futuro su implementación en forma de plugin (por la necesidad de integrarse dentro del propio código de DokuWiki). Concretamente las líneas del código que hacen referencia a a esto son:

```
// the next line should be discommented to enable guest users (not authenticated) enter DokuWiki, see also documentation
# if ($this->as->isAuthenticated()) {
```

Además de descomentar las líneas aquí vistas, la ampliación futura requeriría redirigir el actual enlace del botón de Login del formulario de autenticación de DokuWiki hacia la página de autenticación del Proveedor de Identidad.

Para comprobar el funcionamiento de esto se ha hecho el siguiente cambio manualmente en el fichero inc/template.php:

```
//   $out .= html_btn('login',$ID,'',array('do' => 'login', 'sectok' => getSecurityToken()));

$as = new SimpleSAML_Auth_Simple('default-sp');

$link_as = $as->getLoginURL();

$out .= '<form class="button btn_login" method="post" action="' . $link_as . '"><div class="no"><input type="submit" value="Login" class="button" title="Login" /></div></form>'
```

Aunque realizando este cambio se obtiene el funcionamiento mencionado, la ampliación debería tener en cuenta el uso del evento TPL_CONTENT_DISPLAY que permite el post-proceso del código html para sustituir las líneas indicadas a través de un plugin en lugar de la modificación directa del código de DokuWiki manualmente.

Haciendo balance en cuanto a la experiencia con el software libre, como se ha comentado más arriba el principal escollo en el momento de integrar varias soluciones de software libre ha sido encontrar suficiente información en la documentación de los proyectos en cuestión.

En el caso de DokuWiki los problemas han sido menores, quizás porque la documentación del proyecto se basa en su propio software de wiki permitiendo la creación de una gran base de información alrededor del proyecto. Las mayores dificultades se han encontrado en entender si se podía o no hacer un plugin para el caso de los backends de autenticación. Finalmente se ha visto que sí se podía examinando el ejemplo del plugin de autenticación de facebook *fblogin,* aunque se ha considerado que no valía la pena para nuestro caso (viendo los inconvenientes en el plugin comentado).

En el caso de SimpleSAMLphp se han encontrado más problemas, sobretodo relativos a la funcionalidad del sistema. La documentación de esta aplicación contiene una guía rápida de configuración del sistema, quizás demasiado escueta, que no hace apenas referencia al contexto del protocolo SAML2. Esto provocó que inicialmente no se comprendiera bien cual era el objetivo de la configuración de los diferentes componentes. Una vez, se vio que faltaba algo de información de contexto, sin embargo, se empezó a buscar información en otras fuentes acerca de conceptos como identidad federada, Web Single Sign-On, y el propio protocolo SAML2.

En cuanto a otros componentes utilizados como el servidor Apache o el módulo PHP, no se han encontrado más problemas que los de refresco de algunos conceptos, ya que ya se había tenido alguna experiencia con ellos en el pasado.

Llegamos a las conclusiones finales. La realización de este proyecto tiene una valoración personal generalmente positiva, sobretodo en lo que corresponde a la ejecución del mismo, puesto que en mi opinión se ha ajustado bastante a lo que se esperaba de él, es decir al análisis y planificación hechos inicialmente.

Se ha adquirido experiencia en diferentes áreas como son: la lectura de la documentación, la instalación, configuración e integración de componentes, la gestión del tiempo y la planificación. Por otro lado se han adquirido conocimientos en lo que se refiere a conceptos de autenticación y seguridad, así como en la comprensión y aplicación de las licencias de software libre.

Como punto menos favorable destaco el hecho de que el análisis y la redacción de la documentación, en mi opinión, ha ocupado excesivo tiempo en relación a la implementación. Aunque comprendo que esto sea así normalmente no me ha parecido que la proporción de tiempo empleado en estas tareas haya sido muy equilibrada teniendo en cuenta que el área Web/E-Commerce a la que pertenece el proyecto presuponía un enfoque eminentemente práctico.

Por último, en el contexto más académico, creo que el proyecto ha encajado perfectamente dentro del Máster de Software Libre, tanto por el esfuerzo realizado (número de horas empleadas), como por los roles desempeñados y por las áreas involucradas en el mismo, y también especialmente por la formación específica en el desarrollo de software libre. Es por ello que considero que la realización de este proyecto ha sido de gran valor en cuanto al aprendizaje general, y en particular, en el desarrollo de las habilidades adquiridas entorno al software libre a lo largo del máster.

8. Anexo. Documentación de usuario

Esta documentación de usuario comprende aquellos pasos que se deben realizar para instalar y configurar correctamente el módulo que proporciona integración con SAML2 con la aplicación DokuWiki, haciendo una breve anotación sobre la configuración de SimpleSAMLphp para la correcta integración con DokuWiki. Sin embargo, no es objetivo de esta guía de usuario la instalación y configuración de la plataforma SimpleSAMLphp. Para la implantación de estos servicios se debería consultar la documentación del proyecto donde se pueden encontrar soluciones para múltiples necesidades y variedad de entornos.

La presente documentación está dirigida exclusivamente para administradores de la plataforma DokuWiki, ya que en ella se explican procedimientos que solo puede realizar un usuario con privilegios administrativos, de hecho serán necesarios permisos también a nivel de sistema de ficheros.

La instalación del módulo de autenticación consta de pocos pasos, sin embargo para poder proceder a ella deberemos tener permisos de escritura suficientes en el directorio <ruta raíz dokuwiki>/inc/auth/

En primer lugar deberemos descomprimir el fichero *.zip. Una vez descomprimido copiaremos el fichero ssp.class.php sobre el directorio anteriormente señalado <ruta raíz dokuwiki>/inc/auth/

La configuración de DokuWiki se limitará a cambiar el backend de autenticación actual al del nuevo backend recién instalado. Esto se puede realizar de varias maneras, desde la interfaz gráfica del administrador o directamente cambiando sobre el fichero de configuración de DokuWiki local.php la asignación del array de configuración conf['authtype'] al valor 'ssp', de forma que quede de la siguiente manera:

$conf['authtype'] = 'ssp';

También se debe comprobar que la siguiente línea tenga asignado el valor '1' para tener activo el sistema ACL:

$conf['useacl'] = 1;

Dependiendo de la configuración de la información de autenticación servida por el IdP, esta podría llegar con unos nombres de atributos no esperados. Se recomienda el uso de los siguientes nombres de atributos a configurar en la instancia del IdP relacionada con nuestro SP:

Atributo de DokuWiki	**Nombre del atributo en SimpleSAMLphp**
Identificador de usuario ($USERINFO['user'])	uid
Nombre de usuario ($USERINFO['name'])	cn
Dirección de mail ($USERINFO['mail'])	email
Grupos a los que pertenece el usuario ($USERINFO['grps'])	eduPersonAffiliation

De configurar de la manera anterior los datos de autenticación del IdP se deberán añadir las siguientes líneas en el fichero local.php de la instalación de DokuWiki:

```
$conf['ssp_attr_name'] = 'cn';
$conf['ssp_attr_user'] = 'uid';
$conf['ssp_attr_mail'] = 'email';
$conf['ssp_attr_grps'] = 'eduPersonAffiliation';
```

Para cualquier otra asignación de nombres de atributos se deberán configurar correspondientemente las variables de configuración mostradas.

Finalmente, como se ha dicho al principio, hay que hacer una anotación sobre la configuración de

SimpleSAMLphp necesaria para que este puede funcionar integrado con DokuWiki, ya que por defecto la información de sesión no puede ser compartida por ambos, de forma que se pierde la información de sesión de simpleSAMLphp. Para arreglar esto hay que hacer una ligera modificación sobre uno de los 2 o ambos. La primera solución es más sencilla aunque tiene unos requerimientos de software adicionales:

1) asignar el valor memcache al array 'store.type' => 'memcache' de la configuración de SimpleSAMLphp en *config.php.* (Requisitos: tener instalados los paquetes memcached y php5-memcache en un sistema tipo Debian)

2) asignar el nombre 'DokuWiki' a la cookie de sesión en el mismo fichero de configuración:

'session.phpsession.cookiename' => 'DokuWiki',

y comentar las líneas que asignan los parámertros a la cookie en el fichero init.php de DokuWiki:
session_name("DokuWiki");

```
if (version_compare(PHP_VERSION, '5.2.0', '>')) {
    //session_set_cookie_params(0,DOKU_REL,'',($conf['securecookie'] && is_ssl()),true);
}else{

    //session_set_cookie_params(0,DOKU_REL,'',($conf['securecookie'] && is_ssl()));
```

Configuración extra. (Opcional)

Para permitir el acceso limitado a DokuWiki sin requerir la introducción de usuario y contraseña descomentar las líneas iniciads por '#' en el fichero ssp.class.php.

De realizar esta modificación también será necesaria la edición del fichero inc/template.php para modificar el comportamiento del botón de login redirigiendolo al formulario de login del IdP:

```
//  $out .= html_btn('login',$ID,'',array('do' => 'login', 'sectok' => getSecurityToken()));
$as = new SimpleSAML_Auth_Simple('default-sp');
$link_as = $as->getLoginURL();
$out .= '<form class="button btn_login" method="post" action="' . $link_as . '"><div class="no"><input type="submit" value="Login" class="button" title="Login" /></div></form>';
```

i http://en.wikipedia.org/wiki/Single_sign-on
ii http://en.wikipedia.org/wiki/Wiki
iii http://www.dokuwiki.org/es:syntax
iv http://www.dokuwiki.org/es:dokuwiki
v http://www.dokuwiki.org/plugins
vi http://www.dokuwiki.org/security
vii http://simplesamlphp.org
viii http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language
ix http://www.rediris.es/sir/beneficios.html
x http://www.opensource.org/docs/osdhttp://www.opensource.org/docs/osd
xi http://www.gnu.org/licenses/license-list.html
xii http://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses
xiii http://www.gnu.org/licenses/gpl-faq.html#GPLAndPlugins
xiv http://www.gnu.org/licenses/gpl-faq.html#GPLIncompatibleLibs
xv http://www.gnu.org/licenses/gpl-faq.html#IfInterpreterIsGPL
xvi http://news.netcraft.com/archives/2011/03/09/march-2011-web-server-survey.html
xvii http://simplesamlphp.org/docs/1.8/simplesamlphp-sp-api
xviii http://www.dokuwiki.org/devel:plugin_file_structure
xix http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf
xx http://en.wikipedia.org/wiki/SAML_2.0#Web_Browser_SSO_Profile
xxi https://openidp.feide.no/simplesaml/module.php/metaedit/index.php
xxii http://simplesamlphp.org/docs/1.8/simplesamlphp-idp#section_2
xxiii http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design
xxiv http://www.dokuwiki.org/devel:unittesting
xxv http://simplesamlphp.org/docs/1.8/simplesamlphp-install#section_4
xxvi http://www.dokuwiki.org/devel:authentication_backends
xxvii http://www.dokuwiki.org/devel:authentication_backends#trustexternal