

Índice de contenido

1	Introducción.....	3
1.1	Identificación del trabajo.....	3
1.2	Organización de la documentación.....	3
1.3	Marco de aplicación.....	4
1.4	Objetivos.....	5
1.5	Descripción técnica.....	6
2	Metodologías.....	12
2.1	Metodología.....	12
3	Tecnologías empleadas.....	14
3.1	Mercurial y bitbucket.....	14
3.2	Apache.....	14
3.3	MySQL.....	15
3.4	PHP.....	15
3.5	Mediawiki.....	16
3.5.1	Hooking.....	16
3.5.2	Wrapper de la base de datos.....	16
3.5.3	Internacionalización (i18n) y Localización (L10n).....	17
3.6	Latch.....	17
3.6.1	API de Latch.....	17
3.6.2	SDK de Latch para PHP.....	17
4	Planificación y presupuesto.....	18
4.1	Planificación estimada.....	18
4.2	Seguimiento temporal.....	21
4.3	Desviaciones sobre la planificación inicial.....	24
4.4	Presupuesto.....	25
4.4.1	Presupuesto de hardware.....	25
4.4.2	Presupuesto de software.....	26
4.4.3	Presupuesto de personal.....	28
4.4.4	Presupuesto de impresión.....	28
4.4.5	Presupuesto total.....	29
4.5	Desviaciones sobre el presupuesto inicial.....	29
5	Problemas encontrados y soluciones tomadas.....	29
6	Conclusiones.....	30
7	Futuras ampliaciones.....	31
7.1	Versión para MAC del instalador/desinstalador.....	31
7.2	Versión para Ms Windows del instalador/desinstalador.....	32
7.3	Añadir operaciones y suboperaciones que se puedan latchear.....	32
7.4	Retrocompatibilidad de la internacionalización.....	32
7.5	Traducción de los mensajes de error.....	32
8	Estructura y contenidos del disco adjunto.....	33
9	Bibliografía.....	33

Índice de ilustraciones

Ilustración 1: Pareado de una cuenta Mediawiki con Latch.....	10
Ilustración 2: Proceso de autenticación en MediaWiki.....	11
Ilustración 3: Enfoque metodológico en cascada.....	13
Ilustración 4: Planificación estimada - duración y fechas estimadas de las tareas.....	19
Ilustración 5: Planificación estimada - diagrama de Gantt.....	20
Ilustración 6: Seguimiento de la duración y fechas de realización de las tareas.....	22
Ilustración 7: Diagrama de Gantt del seguimiento temporal.....	23

Índice de tablas

Tabla 1: Estimación temporal.....	19
Tabla 2: Dedicación temporal.....	21
Tabla 3: Presupuesto y amortización del hardware.....	25
Tabla 4: Presupuesto de software.....	27
Tabla 5: Presupuesto de personal.....	28
Tabla 6: Presupuesto de impresión y encuadernación.....	28
Tabla 7: Presupuesto total.....	29

1 Introducción

1.1 Identificación del trabajo

Título: Extensión de autenticación basada en 2º factor con Latch para MediaWiki

Código: EI14/15-044

Curso: 2015/2016

Área de Linguaxes e Sistemas Informáticos

Departamento de Informática

Universidad de Vigo

1.2 Organización de la documentación

La presente documentación se ha organizado en las siguientes partes:

- Manual de usuario: documento creado para el usuario final. Describe las instrucciones de instalación y el modo de empleo de la extensión.
- Manual técnico: documento creado para los responsables del mantenimiento de la extensión. Especifica la estructura de la extensión y la organización de los directorios y los archivos de código del mismo durante el desarrollo de dicha extensión.
- Memoria: documento donde se muestra el desarrollo de la extensión, incluyendo el proceso integración y traducción y los contratiempos sucedidos durante estas tareas.

1.3 Marco de aplicación

La mayoría de servicios que se ofrecen actualmente están conectados a Internet en su

inmensa mayoría, servicios como Play StationNetwork, Xbox Live, Nespresso Club, Correos, Amazon, eBay, etc.

Hoy en día, son cada vez más frecuentes los robos de credenciales online, lo que supone una pérdida económica y de confianza por parte del cliente final. Este robo de identidades está además evolucionando hacia métodos más sofisticados, dificultando más si cabe que los usuarios puedan protegerse.

Una posible solución a esta problemática son los llamados segundos factores de autenticación, que añaden una capa extra de seguridad a mayores de la contraseña. Existen factores de autenticación físicos que pueden ser llevados en el bolsillo o en el llavero como los tokens criptográficos, que son dispositivos electrónicos que almacenan claves criptográficas, como firmas digitales o datos biométricos del usuario.

Los tokens criptográficos son un dispositivo físico, por lo que tienen la desventaja de que se pueden perder con facilidad u olvidarlos en casa y, como consecuencia, hay que recurrir a invalidar el token anterior y adquirir uno nuevo.

Por otra parte, existen también otras soluciones, como tarjetas de coordenadas, usadas comúnmente por entidades bancarias, para que el cliente realice operaciones que impliquen movimiento de fondos o contratación de productos con el fin de que, si sus datos de acceso han sido comprometidos, exista un segundo factor para comprobar la identidad del usuario.

En el mercado están disponibles otro tipo de segundos factores de autenticación, como Latch, que es una suerte de cerrojo digital que el usuario controla con una aplicación desde su smartphone. Con esta aplicación, el usuario final puede poner el cerrojo a las aplicaciones que tenga en su wallet en el teléfono.

Cerrar un latch de una cuenta de usuario para una aplicación, como por ejemplo tuenti, implica que si alguien intenta acceder a esta cuenta cuando el cerrojo está puesto, el atacante no solo no tendrá acceso a la cuenta, sino que además llegará una notificación

desde la aplicación del móvil avisando al usuario legítimo del intento de acceso.

Existen multitud de aplicaciones que ya son compatibles con Latch hoy en día, como por ejemplo Windows enterprise edition, Windows personal edition, Unix, FreeBSD, Linux, Mac, OpenBSD, OpenLDAP, .net, .netNuke, Drupal6, Drupal7, OpenVPN, ownCloud, PHPBB, PHPMyadmin, PrestaShop, Redmine, RoundCube, Squirrelmail, SSH, SugarCRM, Wordpress, Joomla, Moodle y OpenXchange.

También hay servicios web donde puedes poner un pestillo digital, como, Acens, Tuenti, la tienda virtual de Oxxword, Banca Cajamar, Movistar, Universidad internacional de la Rioja y Universidad de Salamanca.

Todas estas aplicaciones y servicios web han sido adaptados para soportar esta nueva tecnología. Este proyecto consiste en el desarrollo de un plugin para integrar Latch con la plataforma MediaWiki, un software libre para crear wikis usado por Wikipedia y otros proyectos de la fundación Wikimedia.

1.4 Objetivos

A raíz de los problemas citados en el apartado anterior, se ha considerado útil integrar Latch en Mediawiki para dar la posibilidad a los usuarios de este tipo de wikis de utilizar un segundo factor de autenticación y añadir así una capa extra de seguridad a sus cuentas.

Como objetivo principal se quiere desarrollar una extensión de autenticación para MediaWiki basada en Latch. Para cumplir este objetivo, se deberán cumplir los siguientes subobjetivos parciales:

- El usuario puede autenticarse en su perfil de MediaWiki y, una vez autenticado, tiene una opción en un formulario de opciones de configuración dentro de Mediawiki donde puede solicitar el pareo de su cuenta de MediaWiki con Latch. Este pareo se realiza contra los servidores de Latch y no se envía ningún dato sobre el tipo de cuenta o nombre de usuario, tan solo un token numérico OTP

(One Time Password) que se envía a la app de Latch en el teléfono móvil cuando el usuario realiza el pareado de la cuenta de MediaWiki con la cuenta de Latch.

Solo es necesario realizar el pareado de la cuenta una única vez; una vez que la cuenta esté pareada el usuario puede utilizar la aplicación del teléfono móvil para gestionar el Latch de MediaWiki cuando así lo desee.

- Aquellos usuarios con cuenta latcheada solo pueden hacer login cuando el latch esté abierto, los intentos de acceso con el latch cerrado son notificados a través de la aplicación en el móvil.
- La extensión sigue los estándares de MediaWiki de forma que sea fácil su inclusión y mantenimiento en el proyecto de MediaWiki.

Es preciso destacar que la aplicación móvil proporcionada por Latch, externa a este proyecto, permite por su parte, la gestión de la cuenta de MediaWiki así como de otras cuentas y servicios que el usuario quiera latchear. En esencia, el usuario puede abrir un pestillo, cerrar un pestillo, activar y desactivar el autobloqueo por tiempo, por ejemplo se puede habilitar un "modo noche" que bloquee automáticamente el acceso en el horario nocturno definido por el usuario en las cuentas que el usuario configure.

1.5 Descripción técnica

Para el desarrollo de esta integración de Latch en Mediawiki se ha usado PHP 5.5.20 con los módulos core.c, mod_so.c y http_core.c.

La versión del servidor MySQL usado para el proyecto es 5.6.22 MySQL Community Server (GPL). Se ha integrado en una instalación de MediaWiki versión 1.23.8. Todo ello sobre una máquina con Fedora release 19 (Schrödinger's Cat).

Se trata de una extensión escrita en PHP utilizando hooks contra Mediawiki, y basada en el patrón de arquitectura de software conocido como modelo-vista-controlador (MVC), que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Para ello se han construido tres componentes distintos, que son, el modelo, la vista y el controlador. Es decir, por un lado, se han definido componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

- El controlador es el módulo contenido en el archivo `LatchController.php` y se encarga de la entrada, que en el caso de este proyecto, son las funciones relacionadas con el pareo de la cuenta, el despareo de la cuenta y la comprobación del estado del latch activo actualmente.
- La vista es el módulo que hace referencia al archivo `LatchAccount.php` y se encarga de renderizar la vista de pareo para el usuario que está despareado y la vista de despareo si el usuario está pareado.
- El modelo está contenido en el archivo `DBhelper.php` y se encarga de interactuar con la base de datos de Mediawiki para almacenar, recoger o borrar datos de la misma y proveer de esta información a la vista y el controlador. Es decir, se trata de un módulo intermedio entre el modelo y la vista que responde a eventos (acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información.

Para realizar la integración de Latch con Mediawiki se ha utilizado el SDK de PHP que provee Eleven Paths (empresa propietaria del producto Latch) para utilizar la API de Latch con mayor sencillez.

Este SDK se encuentra almacenado en el directorio `PHP_SDK` que contiene los archivos `Latch.php`, `LatchResponse.php` y `Error.php`

El archivo `LatchConfig.php` contiene la configuración necesaria para enviar consultas al servidor de Latch e identificar unívocamente a la persona que está usando la aplicación

a través de su cuenta de developer de Eleven Paths.

Para poder implantar estas funcionalidades, se ha utilizado un editor de texto simple llamado Geany para la codificación del código PHP, el navegador Firefox en su versión 34.0 para Linux para testear el código sobre Mediawiki, y el terminal de linux para acceder al SGBD MySQL mediante comandos de consola.

La arquitectura del sistema está compuesta por cuatro agentes principales:

- **Servicio MediaWiki.** Configurado sobre un servidor Apache en una máquina linux.
- **Núcleo Latch.** El núcleo esencial del servicio que acepta y procesa las solicitudes de estado enviadas desde el proveedor de servicios y bloquea/desbloquea la cuenta de usuario en función del estado definido por el usuario. También es responsable de enviar los tokens de contraseña de un solo uso (OTP).
- **API y SDK de Latch.** Latch ofrece SDKs y API del servicio web para los lenguajes de programación más utilizados, en este caso se ha usado el SDK de PHP.
- **App móvil Latch.** Es una aplicación que permite a los usuarios iniciar sesión en el servicio Latch para administrar, desde sus dispositivos móviles, el estado de los servicios que han pareado. Pueden bloquear/desbloquear su cuenta con facilidad, programar períodos de bloqueo de su cuenta, activar el segundo factor de autenticación, obtener contraseñas de un solo uso, recibir alertas y supervisar el acceso a sus cuentas.

En el presente proyecto, se ha intervenido en MediaWiki desarrollando un nuevo plugin con la ayuda del SDK de Latch para PHP que el usuario final maneja desde la aplicación para el móvil.

Inicialmente el usuario ha de tener una cuenta de desarrollador en Latch, una cuenta en MediaWiki y la aplicación de Latch instalada en el teléfono móvil.

Para la creación de un latch asociado a una cuenta de usuario en MediaWiki se ha de

realizar el siguiente proceso:

1. El usuario ha de logearse en MediaWiki. Una vez en su cuenta, deberá introducir un código OTP en el formulario para parear su cuenta de Mediawiki con Latch.
2. En la aplicación del móvil existe la opción "añadir servicio", que lleva a una pantalla con la posibilidad de "generar código de pareado" (One Time Password).

La aplicación móvil envía entonces una petición al servidor de Latch que devuelve un token temporal de pareado que caduca pasados 60 segundos.

3. El usuario introduce el OTP en la web de MediaWiki y su cuenta queda así pareada con el servicio de Latch.

Solo es necesario introducir el token temporal una única vez por cada identidad.

4. Una vez introducido el token temporal se envía una petición de pareado al server de Latch.

El server de Latch devuelve al plugin de MediaWiki el código identificador del nuevo latch (application ID) y un código (secret), el cual será guardado en la BD de MediaWiki y empleado para consultar el estado de este nuevo cerrojo en futuros accesos a la cuenta de Mediawiki.

Durante todo este proceso no se ha transmitido ningún dato personal (para crear una cuenta en Latch solo hace falta un nombre de usuario y una cuenta de correo) ni el terminal del móvil almacena ninguna información relativa a las cuentas que protege, el servidor de Latch no conoce con que identidad está pareado cada Latch debido a la implementación del servicio utilizando tokens temporales de pareado para cada identidad digital. La siguiente figura muestra de forma gráfica el proceso anteriormente descrito.

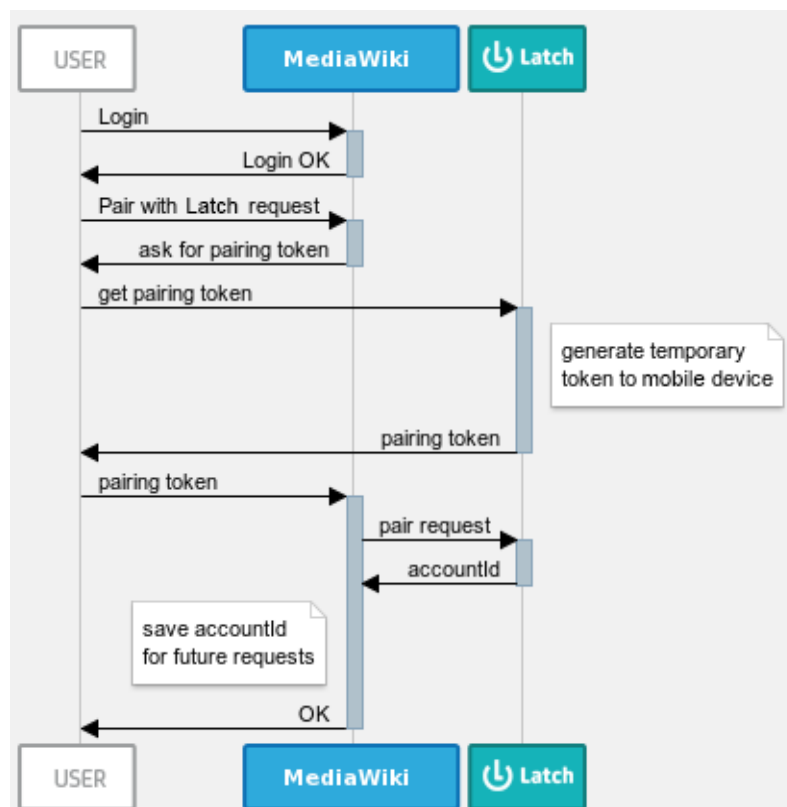


Ilustración 1: Pareado de una cuenta Mediawiki con Latch.

Fuente: documentación del desarrollador de Latch.

Finalmente, el usuario tien una lista de identidades digitales que puede activar o desactivar desde la aplicación del móvil como si de un mando a distancia se tratase.

El proceso de verificación de si el pestillo digital de una cuenta en concreto está ON/OFF lo hace el servidor de MediaWiki por medio de la API de Latch y envía una consulta de estado asociado al pestillo digital del usuario.

El servidor contestará ON/OFF y, en caso de estar OFF, el servidor bloqueará el acceso y el usuario recibirá una alerta que le indica que alguien ha usado su usuario y contraseña correctos pero que el acceso ha sido bloqueado, ayudando a detectar el robo de identidad en tiempo real.

La siguiente figura muestra el proceso de autenticación en MediaWiki una vez integrado con Latch.

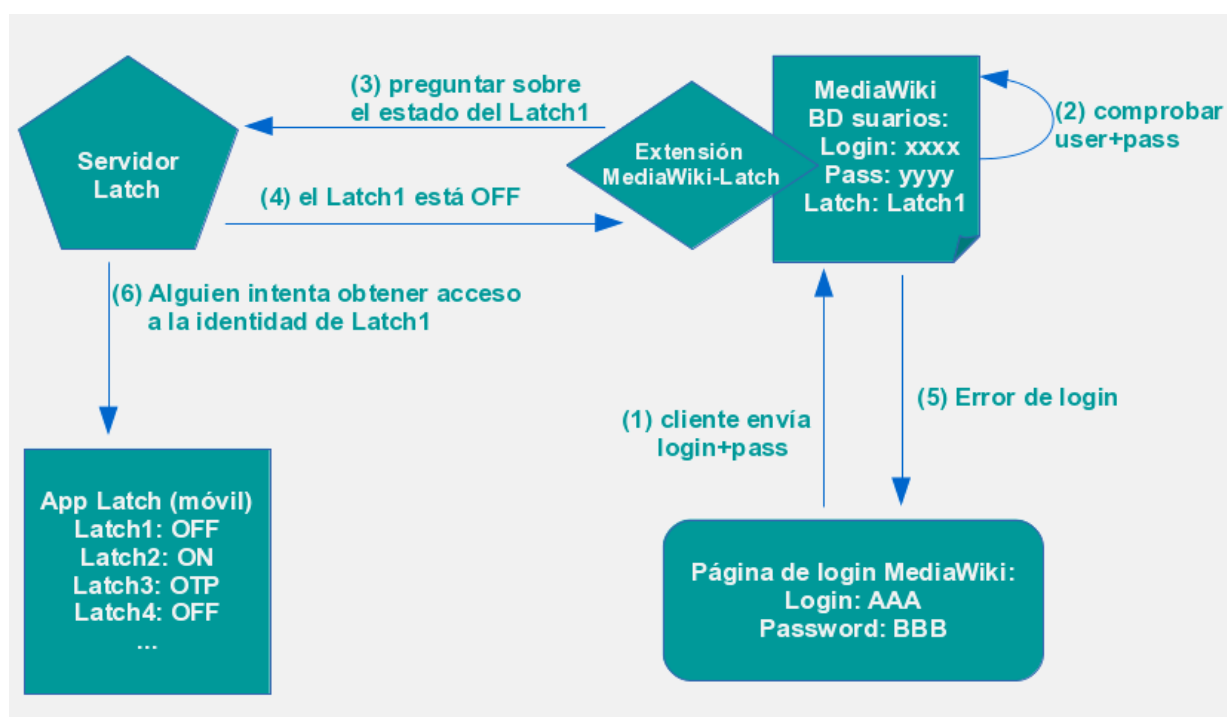


Ilustración 2: Proceso de autenticación en MediaWiki una vez extendido con Latch.

2 Metodologías

Para elaborar el plugin que conecta la plataforma MediaWiki con el sistema Latch, descrito anteriormente, se ha usado un proceso de desarrollo de software denominado modelo en cascada.

2.1 Metodología

El modelo de cascada, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

El proyecto se ha desarrollado en seis etapas: inicio, análisis, diseño, implementación, integración y pruebas, documentación.

Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase.

El modelo en cascada puede ser aplicado a este proyecto debido a que al desarrollar un plugin entre dos plataformas ya existentes, las fases del proyecto están muy claramente definidas y es viable seguir el flujo secuencial que propone el modelo.

- En la primera fase de inicio se ha realizado un estudio previo de los sistemas a integrar.
- En la fase de análisis, se ha descrito el plugin a partir de los requisitos previos.
Esto viene dado por los sistemas externos (MediaWiki y Latch) con los que trabaja el proyecto.
- En la fase de diseño se ha diseñado la arquitectura que permite poner en marcha todo el sistema.
- En la siguiente etapa, la fase de implementación, se ha construido el software a

partir de la arquitectura diseñada y los requisitos capturados en la etapa de análisis de requisitos.

- En la etapa de integración y pruebas se han llevado a cabo las pruebas necesarias para verificar el correcto funcionamiento del sistema y su integración con las plataformas.
- Cada una de estas fases lleva consigo una fase de documentación de la misma.

Una de las ventajas del modelo de desarrollo en cascada es que es un modelo fácil de implementar y entender.

Este modelo está orientado a documentos y es un modelo conocido y utilizado con frecuencia que promueve una metodología de trabajo efectiva: definir antes que diseñar, diseñar antes que codificar.

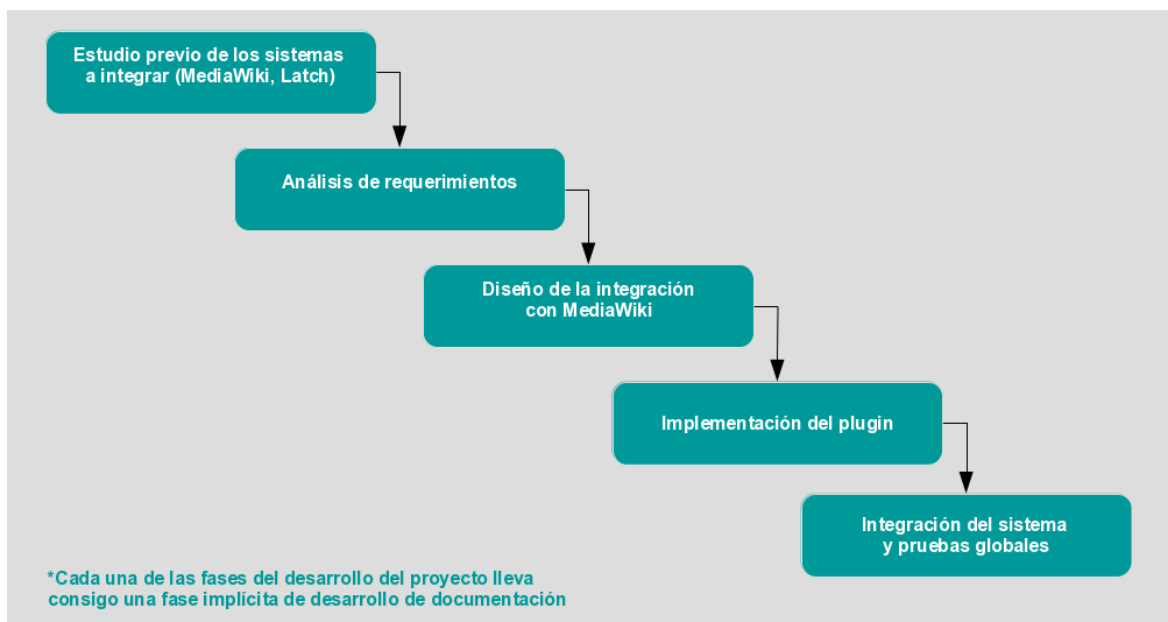


Ilustración 3: Enfoque metodológico en cascada.

3 Tecnologías empleadas

3.1 Mercurial y bitbucket

Mercurial es un sistema de control de versiones multiplataforma, para desarrolladores de software. Está implementado principalmente en Python. Fue escrito originalmente para funcionar sobre GNU/Linux, pero ha sido adaptado para Windows, Mac OS X y otro sistemas tipo Unix.

Las principales metas de desarrollo de Mercurial incluyen un gran rendimiento y escalabilidad; desarrollo completamente distribuido, sin necesidad de un servidor; gestión robusta de archivos tanto de texto como de binarios; y capacidades avanzadas de ramificación e integración, todo ello manteniendo sencillez conceptual.

El código fuente se encuentra disponible bajo los términos de la licencia GNU GPL versión 2 lo que clasifica a Mercurial como software libre.

Bitbucket es una plataforma de desarrollo colaborativo para alojar proyectos usando el sistema de control de versiones Mercurial o Git.

Bitbucket ofrece planes comerciales y gratuitos. Se ofrecen cuentas gratuitas con un número ilimitado de repositorios privados.

El servicio está escrito en Python.

3.2 Apache

Apache es un servidor web HTTP de código abierto para plataformas Unix, Windows, Macintosh que implementa el protocolo HTTP/1.12.

Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado.

3.3 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

MySQL es un producto de software libre con un esquema de licenciamiento dual .

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

MySQL es muy utilizado en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla.

Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

3.4 PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor, originalmente diseñado para el desarrollo web de contenido dinámico.

Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos.

El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

PHP ha evolucionado, por lo que ahora incluye también una interfaz de línea de comandos, que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas

operativos y plataformas sin ningún costo.

Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP.

3.5 Mediawiki

MediaWiki es un software para wikis libre, programado en el lenguaje PHP.

Es el software usado por Wikipédia y otros proyectos de la Fundación Wikimedia (Wikcionario, Wikilibros, etc.)

Ha tenido una gran expansión desde el año 2005, existiendo un gran número de wikis basadas en este software que no mantienen relación con dicha fundación, aunque sí comparten la idea de la generación de contenidos de manera colaborativa.

Se encuentra bajo la licencia de software GNU, General Public License.

3.5.1 Hooking

Los **hooks** permiten que se ejecute código propio cuando ocurre un evento definido (como por ejemplo guardar una página de la wiki, o que un usuario haga login).

Esto permite definir código propio que se ejecuta en ciertos puntos del código principal de Mediawiki.

Los hooks permiten mantener las líneas principales del código del core de Mediawiki limpio, claro y fácil de leer y entender, de este modo es más sencillo escribir extensiones para la plataforma.

3.5.2 Wrapper de la base de datos

Mediawiki provee de una capa de abstracción hacia la base de datos, de modo que, a no ser que se esté trabajando en la capa de abstracción, no es buena práctica llamar directamente a funciones en PHP de la base de datos.

La razón por la cual es importante usar métodos de alto nivel en lugar de construir consultas propias, es asegurarse de que el código funcionará correctamente sin importar el tipo de base de datos que utilice Mediawiki.

Este sistema permite que Mediawiki soporte bases de datos de diversos fabricantes (`MySQL`, `SQLite`, `PostgreSQL`, `Oracle` y `DB2`) evitando que el código fuente tenga que ser específico para cada una de ellas.

3.5.3 Internacionalización (i18n) y Localización (L10n)

La internacionalización y la localización son medios para adaptar software a los diferentes idiomas, diferencias regionales y requerimientos técnicos.

En otras palabras, `i18n` y `L10n` permiten dar soporte y satisfacer las necesidades de múltiples localizaciones geográficas con distintos idiomas o con diferentes variantes del mismo idioma. Mediawiki implementa estos sistemas a través del objeto `Language` que se encuentra definido en el fichero `Language.php` y los ficheros de mensajes `.json`.

3.6 Latch

3.6.1 API de Latch

Para realizar la autenticación en el servicio Latch, todas las solicitudes a la API de Latch deben estar firmadas. El proceso de firma es una versión simplificada del protocolo OAuth de dos vías. Cada solicitud HTTP a la API debe ir acompañada de dos encabezados de autenticación: `Authorization` y `Date`.

3.6.2 SDK de Latch para PHP

El SDK de Latch para PHP está compuesto por un conjunto de clases que permiten a los desarrolladores integrar Latch en sus aplicaciones.

El SDK proporcionado por Eleven Paths, es software libre, que puede ser redistribuido y modificado bajo los términos de la GNU Lesser General Public License.

4 Planificación y presupuesto

4.1 Planificación estimada

En esta sección se presenta la planificación del proyecto que ha sido llevada a cabo antes de la realización del proyecto.

Se estima una duración total de 60 días, trabajando durante 5 horas al día, 7 días a la semana. Por tanto, en cada semana se emplean 35 horas para la elaboración del proyecto, esto implica una duración de proyecto de 8,5 semanas.

El comienzo del proyecto será el día 30 de marzo de 2015, por lo que, según la estimación anterior, la fecha de finalización del mismo será el 20 de mayo de ese mismo año. En resumen, trabajando 5 horas al día durante 60 días, el proyecto tendrá una duración estimada de 300 horas.

El desarrollo del proyecto se realizará en 3 fases:

- Inicio
- Desarrollo del plugin
- Documentación

Las siguientes tablas muestran tanto la planificación en horas por semana de cada una de las fases del proyecto como la planificación con las fechas de inicio y fin de cada una de las fases.

Fase		Dedicación temporal	
		Días	Semanas
Estudio previo	Elaboración arquitectura: casos de uso críticos	3	3,7
	Estudio del SDK para PHP de Latch	7	
	Estudio del sistema de integración de plugins en Mediawiki	8	
	Análisis del funcionamiento de la API de Latch	8	
Desarrollo del plugin	Análisis de requerimientos: estudio de Latch y Mediawiki	5	5,6
	Diseño de la integración con Mediawiki	10	
	Implementación	12	
	Integración	6	
	Pruebas globales	6	
Documentación	Manuales de usuario	40	5,7
Total proyecto		~65	~8,5

Tabla 1: Estimación temporal; dedicación de 35 horas por semana (5 horas/día, 7 días/semana)
















	Nombre	Duración	Inicio	Fin
	 Extensión de autenticación con Latch para MediaWiki	60.8d	30/03/2015	20/05/2015
	 Inicio	22.4d	30/03/2015	16/04/2015
	 Estudio previo	22.4d	30/03/2015	16/04/2015
	Elaboración arquitectura: casos de uso críticos	3d	30/03/2015	31/03/2015
	Estudiar el SDK para PHP de Latch	7d	31/03/2015	06/04/2015
	Estudiar el sistema de integración de plugins en MediaWiki	8d	06/04/2015	10/04/2015
	Analizar el funcionamiento de la API de Latch	8d	10/04/2015	16/04/2015
	 Desarrollo del plugin	36.4d	17/04/2015	19/05/2015
	Análisis de requerimientos: estudio de Latch, Mediawiki	5d	17/04/2015	22/04/2015
	Diseño de la integración con MediaWiki	10d	22/04/2015	30/04/2015
	Implementación	12d	30/04/2015	11/05/2015
	Integración	6d	11/05/2015	14/05/2015
	Pruebas globales	6d	14/05/2015	19/05/2015
	 Documentación	40d	16/04/2015	20/05/2015
	Manuales de usuario	40d	16/04/2015	20/05/2015

Ilustración 4: Planificación estimada - duración y fechas estimadas de las tareas

Extensión de autenticación basada en 2º factor con Latch para MediaWiki

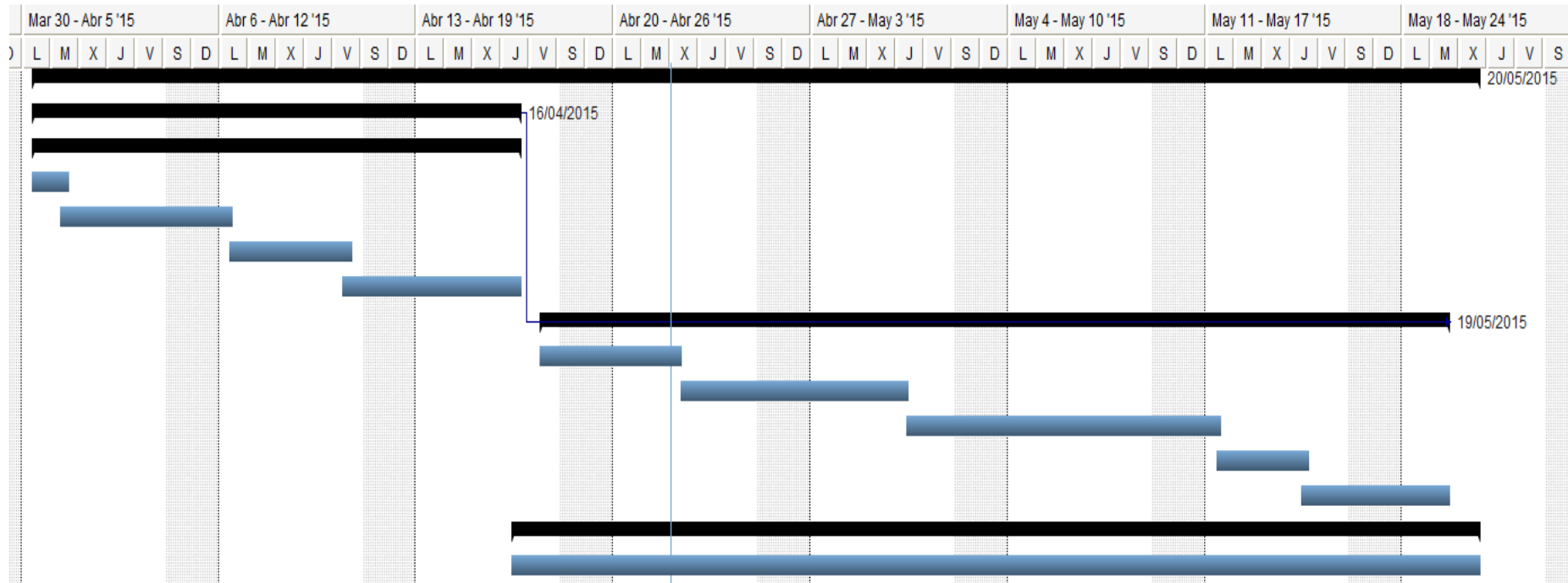


Ilustración 5: Planificación estimada - diagrama de Gantt

4.2 Seguimiento temporal

Este apartado muestra la duración de cada una de las fases del proyecto. Se muestra el tiempo total real que ha sido invertido en la realización del proyecto. La tabla 2 detalla la duración en días y semanas de cada una de las fases del proyecto.

Fase		Dedicación temporal	
		Días	Semanas
Estudio previo	Elaboración arquitectura: casos de uso críticos	7	4,5
	Estudio del SDK para PHP de Latch	7	
	Estudio del sistema de integración de plugins en Mediawiki	13	
	Análisis del funcionamiento de la API de Latch	5	
Desarrollo del plugin	Análisis de requerimientos: estudio de Latch y Mediawiki	8	5,2
	Diseño de la integración con Mediawiki	10	
	Implementación	10	
	Integración	6	
	Pruebas globales	3	
Documentación	Manuales de usuario	33	4,7
Total proyecto		~65	~8,5

Tabla 2: Dedicación temporal

La ilustración 6 muestra la duración y las fechas de realización de cada una de las fases del proyecto.












		Nombre	Duración	Inicio	Fin
1		☐ Extensión de autenticación con Latch para MediaWiki	60.6d	03/08/2015	23/09/2015
2		☐ Inicio	60.6d	03/08/2015	23/09/2015
3		☐ Estudio previo	27.4d	03/08/2015	26/08/2015
4		Elaboración arquitectura: casos de uso criticos	7d	03/08/2015	07/08/2015
5		Estudio del SDK para PHP de Latch	4d	07/08/2015	11/08/2015
6		Estudio del sistema de integración de plugins en Mediawiki	13d	11/08/2015	21/08/2015
7		Análisis del funcionamiento de la API de Latch	5d	21/08/2015	26/08/2015
8		☐ Desarrollo del plugin	33.4d	26/08/2015	23/09/2015
9		Análisis de requerimientos: estudio de Latch y Mediawiki	8d	26/08/2015	01/09/2015
10		Diseño de la integración con Mediawiki	10d	01/09/2015	09/09/2015
11		Implementación	10d	09/09/2015	17/09/2015
12		Integración	6d	17/09/2015	22/09/2015
13		Pruebas globales	3d	22/09/2015	23/09/2015
14		☐ Documentación	33d	26/08/2015	23/09/2015
15		Manuales de usuario	33d	26/08/2015	23/09/2015

Ilustración 6: Seguimiento de la duración y fechas de realización de las tareas

Extensión de autenticación basada en 2º factor con Latch para MediaWiki

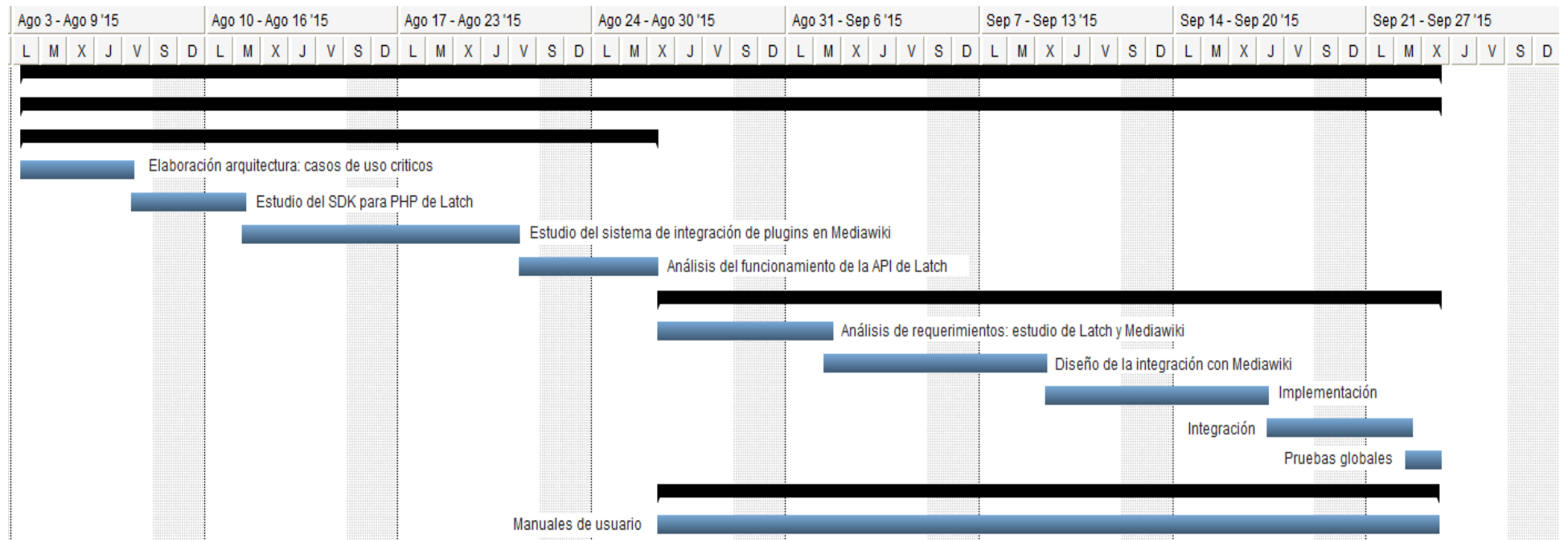


Ilustración 7: Diagrama de Gantt del seguimiento temporal

4.3 Desviaciones sobre la planificación inicial

Como se puede observar al comparar ambas tablas existe un desajuste en el seguimiento sobre la planificación planteada al principio del proyecto.

El retraso en el inicio se debe a una planificación donde no se tuvieron en cuenta los exámenes parciales, prácticas y trabajos de las asignaturas cursadas durante ese período por lo cual se retrasó el inicio del proyecto hasta principios de agosto.

Como se observa al comparar ambas tablas existe un desajuste en el seguimiento de 14 días sobre la planificación planteada al inicio del proyecto. Analizando el porqué de ese atraso en los plazos se detectan dos razones.

El principal motivo es una estimación demasiado optimista donde la mayoría de las tareas han tenido una duración mayor de lo esperado. Bien sea por desconocimiento de la plataforma o por una productividad menor de la esperada.

Otra razón de peso se puede comprobar en los diagramas de Gantt, durante los meses de abril y mayo han existido numerosas retrasos en completar las tareas asignadas. Todas estas paradas fueron causadas por exámenes y trabajos destinados a asignaturas cursadas durante el presente año.

Sin embargo la duración total en semanas que se planificó inicialmente coincide con precisión bastante aproximada con la el tiempo real empleado en el desarrollo del proyecto.

No obstante la fase inicial de estudio de las tecnologías a emplear se desvía en una semana de planificación inicial que era demasiado optimista.

4.4 Presupuesto

Para llevar a cabo el desarrollo de este proyecto se ha empleado software libre siempre que ha sido posible, para evitar depender de licencias, evitar gastos innecesarios y poder distribuir el software creado con el mínimo coste monetario posible.

En las siguientes tablas se recoge el presupuesto del proyecto, teniendo en cuenta las siguientes consideraciones:

- El tiempo total del proyecto han sido 8,5 semanas
- La vida útil de un portátil son 4 años (48 meses)
- La duración del proyecto han sido 2,12 meses.

El presupuesto se ha organizado en compras de hardware, software, presupuesto de personal y previsión de gasto de material fungible.

4.4.1 Presupuesto de hardware

En la tabla 3 se describe el hardware que se ha empleado para la realización de este proyecto, el costo y la amortización del mismo.

Hardware	Descripción	Coste	Amortización
Ordenador portátil	Lenovo Thinkpad X201	1000€	44,17€

Tabla 3: Presupuesto y amortización del hardware

4.4.2 Presupuesto de software

Todo el software utilizado es libre y gratuito por tanto no repercute en el cómputo del coste ni en la amortización ya que el coste total de software es 0€.

La tabla 3, que se muestra a continuación, contiene una lista del software que ha sido usado en la realización del proyecto, también se indica el tipo de licencia de cada uno de los productos y servicios en la nube que se han usado.

SOFTWARE	TIPO	LICENCIA	COSTE
Fedora 19	Sistema operativo	Fedora 19 License Agreement	0€
Apache v2.4.10	Servidor web HTTP	Apache License V2.0	0€
MySQL Community Server v5.6.22	SGBD	GPL	0€
PHP v5.5.20	Lenguaje de programación	PHP License v3.01	0€
MediaWiki v1.23.8	Software para crear wikis	GNU, General Public License.	0€
Latch PHP SDK	SDK	GNU Lesser General Public License V2.1	0€
Latch App para Android v1.4.2	Aplicación para smartphone	GNU Lesser General Public License V2.1	0€
Cuenta desarrollador de Latch	Configuración de cuentas y aplicaciones	Suscripción Community	0€
Dropbox	Sistema de backup online	Digital Millennium Copyright	0€
Geany v1.24.1	IDE	GNU GPL V2	0€
Mercurial	Control de versiones	GPLv2+	0€
Bitbucket	Plataforma de desarrollo colaborativo	Atlassian Community license program	0€
LibreOffice v4.4.5.2	Editor de textos	Licencia Pública Mozilla v2.0	0€
Mozilla 34.0 for Linux	Navegador web	Mozilla Public License 2.0	0€
Google Chrome v45.0	Navegador web	Copyright 2015 Google Inc.	0€
Ganttter	Diagramas de Gantt (web)	Suscripción Community	0€
Cacao	Diagramas UML (web)	Copyright 2004-2015 Nulab Inc, plan gratuito	0€
COSTE TOTAL		0€	

Tabla 4: Presupuesto de software

4.4.3 Presupuesto de personal

La tabla 5 recoge los diferentes roles adoptados a la hora de desarrollar el proyecto, así como el coste por hora y las horas necesarias para la realización del proyecto:

Material	Coste/hora	Horas trabajadas	Coste
Analista	18€	~18	324€
Programador	20€	~20	400€
Documentación	10€	~24	240€
<i>TOTAL</i>	>>	>>	964€

Tabla 5: Presupuesto de personal

4.4.4 Presupuesto de impresión

La tabla 6 muestra el coste de impresión y encuadernación relacionados con el material fungible empleado en el desarrollo de este proyecto:

Concepto	Coste
Impresión	50€
Encuadernación	40€
<i>TOTAL</i>	90€

Tabla 6: Presupuesto de impresión y encuadernación

4.4.5 Presupuesto total

Concepto	Coste
Hardware	44,17€
Software	0€
Personal	964€
Impresión y encuadernación	90€
<i>TOTAL</i>	<i>1098,17€</i>

Tabla 7: Presupuesto total

4.5 Desviaciones sobre el presupuesto inicial

El coste previsto del proyecto resultó ser el mismo que el coste real final, ya que los días trabajados han sido los mismos y no se ha necesitado ninguna inversión a mayores en ninguna de las partes que necesitan inversión económica.

5 Problemas encontrados y soluciones tomadas

Durante la instalación y configuración de Mediawiki se produjo un error que impedía la correcta instalación de Mediawiki en el sistema.

En ese momento no existía información disponible sobre este conflicto en la documentación online de Mediawiki.

Lo que en principio parecía un problema con los permisos de acceso (lectura/escritura) al archivo `LocalSettings.php`, no era tal, si no un conflicto con el módulo de seguridad SELinux del kernel de Linux (SecurityEnhanced Linux) activo por defecto en la distribución Fedora sobre la cual se ha desarrollado el proyecto.

Tras varias pruebas, cambiando el dueño, grupo y los permisos al archivo `LocalSettings.php` se realizó una búsqueda a mayores sobre la naturaleza del problema en cuestión, hallando como solución desactivar el módulo SELinux, tras lo cual, se finalizó con éxito la instalación y configuración de Mediawiki en el sistema.

Actualmente, este conflicto, se encuentra listado en la documentación oficial online de Mediawiki en la siguiente URL: <https://www.mediawiki.org/wiki/SELinux>

6 Conclusiones

La integración de dos sistemas ya existentes requiere adaptarse a los estándares y las buenas prácticas establecidas por la plataforma de desarrolladores de Mediawiki, y del mismo modo es necesario un estudio de la API de Latch.

Cabe destacar también el proceso de internacionalización (i18n) y de localización (L10n) que permite adaptar el software para que pueda ser potencialmente traducido a varios idiomas y regiones sin realizar cambios en la ingeniería del código fuente.

Relativo a conclusiones de índole más personal, indicar que en este aspecto en concreto, me sorprendió la robustez con la que están diseñados los sistemas i18n y L10n en Mediawiki, de modo que tan solo es necesario realizar cambios en los archivos `.json` sin interferir en el core de Mediawiki.

Ha sido gratificante el uso de estas metodologías y técnicas que es posible que sean de utilidad en futuros proyectos.

En lo que respecta al aprendizaje, el uso de tecnologías que desconocía, como son las técnicas de hooking para interceptar llamadas a funciones, eventos y mensajes que se pasan entre componentes de software, me ha resultado interesante y me ha ayudado a comprender que no existe una única solución a un mismo problema y que estas técnicas pueden aplicarse independientemente de la tecnología y la metodología de desarrollo usadas. La realización del proyecto ha resultado interesante aunque también compleja,

debido a que ha sido la primera vez que he afrontado un proyecto de estas características, tanto a nivel de desarrollo como a nivel de planificación y también debido al uso de tecnologías desconocidas a priori.

El aspecto más gratificante ha sido que como resultado de la realización del proyecto cualquiera que desee poner un segundo factor de autenticación a su sistema de wikis con Mediawiki tiene el código disponible para ello.

Por último destacar, que también he aprendido a apreciar la documentación, ya que he tenido que hacer un uso bastante intensivo tanto de la documentación de Latch como de Mediawiki, y me he dado cuenta de que es realmente importante, tanto en proyectos grandes como pequeños, que se documente correctamente, tanto para futuras consultas, como para otros desarrolladores que trabajen en el mismo proyecto.

7 Futuras ampliaciones

Un proyecto de estas dimensiones, asignado a proyecto de fin de grado, al que si comparamos la carga de dedicación con el tiempo concedido para su realización, éste se queda corto. Esto limita el resultado final del mismo dejándolo con la posibilidad de añadirle numerosas mejoras o innovaciones.

A continuación se detallan algunos aspectos del proyecto en los que sería interesante trabajar como posibles ampliaciones del mismo pero que no han sido llevados a cabo por el momento.

7.1 Versión para MAC del instalador/desinstalador

Para hacer más sencilla al usuario la tarea de instalar el plugin en el sistema sería interesante implementar un instalador o un pequeño script para instalar el plugin en sistemas OS X de modo que el usuario final no tenga que buscar las rutas de instalación donde copiar los archivos, crear los directorios necesarios, añadir líneas al fichero LocalSettings.php, crear la tabla en la base de datos o cambiar los permisos a los

directorios pertinentes.

Del mismo modo sería interesante crear un desinstalador para facilitar la tarea de eliminar los archivos si el usuario decide desinstalar el plugin o dejar de usarlo.

7.2 Versión para Ms Windows del instalador/desinstalador

Del mismo modo que se explica en el apartado anterior, sería interesante crear un instalador y un desinstalador para sistemas Ms. Windows para facilitar al usuario final la tarea de instalar y desinstalar el plugin en el sistema.

7.3 Añadir operaciones y suboperaciones que se puedan latchear

Quizás sería interesante añadir operaciones como "editar página de la wiki" o "editar preferencias de usuario" para que el usuario pueda poner un latch sobre cada operación individualmente a mayores de controlar el acceso a la cuenta como un todo, que es la funcionalidad que tiene actualmente el plugin.

7.4 Retrocompatibilidad de la internacionalización

Se podría estudiar la posibilidad de hacer compatibles las traducciones que están hechas para esta extensión a versiones anteriores de Mediawiki donde el sistema usado es diferente.

7.5 Traducción de los mensajes de error

Por falta de tiempo no ha sido posible investigar como llevar a cabo la traducción de los mensajes de error ya que Mediawiki no usa el mismo sistema para estos mensajes que para el resto de mensajes del sistema por tanto actualmente solo están disponibles en la versión en castellano.

8 Estructura y contenidos del disco adjunto

El disco que se distribuye con el proyecto tiene la siguiente estructura de directorios:

- docs: contiene un .txt con la documentación siguiente el estándar de Mediawiki.
- extensions: contiene el directorio Latch, dentro del cual se hayan los archivos con el código fuente de la extensión. También contiene el directorio i18n que contiene lo archivos de internacionalización, y el directorio PHP_SDK que contiene el SDK de PHP para utilizar la API de Latch
- linux_install: contiene los scripts install.sh, uninstall.sh para instalar y desinstalar el plugin en entornos GNU/Linux.

En este directorio se encuentra también el archivo linux_install.txt que explica de forma detallada cómo instalar el plugin en entornos GNU/Linux.

- windows_install: contiene el archivo windows_install.txt que explica de forma detallada como instalar el plugin en entornos Ms. Windows.

También contiene el archivo LatchExtension.txt que contiene la documentación.

9 Bibliografía

Ingeniería del software: Un enfoque práctico. Roger S. Pressman, 2002

The UML reference manual. Rumbaugh, Jacobson, Booch, 1998

Desarrollo web con PHP y MySQL. Luke Welling, Laura Thomson, 1997

La Biblia de PHP 6 y MySQL. Steve Suehring, Tim Converse, Joyce Park, 2009

Documentación del desarrollador de Latch:

https://latch.elevenpaths.com/www/developers/doc_api, 21 octubre 2015

Manual para desarrollar extensiones en MediaWiki:

<https://www.mediawiki.org/wiki/Manual:Extensions>, 21 octubre 2015