

Diseño de una estrategia para la elaboración de manuales de usuario en ambientes
colaborativos y distribuidos

Autor

Freddy Alexander Sandoval Villamizar

Director

Luis Alberto Esteban Villamizar
Magister en Informática

Ingeniería de Sistemas
Departamento de Eléctrica, Electrónica, Sistemas Y Telecomunicaciones
Facultad de Ingeniería y Arquitectura



Universidad de Pamplona
Pamplona, 2017

Índice

1	INTRODUCCIÓN	1
1.1	PLANTEAMIENTO DEL PROBLEMA	2
1.2	OBJETIVOS	3
1.2.1	<i>Objetivo general</i>	3
1.2.2	<i>Objetivos específicos</i>	3
1.3	JUSTIFICACIÓN	4
1.4	METODOLOGÍA	5
2	MARCO TEÓRICO	6
2.1	INGENIERÍA DEL SOFTWARE	6
2.1.1	<i>Estándar 1074</i>	7
2.1.2	<i>Estándar 12207</i>	11
2.1.3	<i>Proceso de documentación</i>	13
2.2	OPEN SOURCE Y SOFTWARE LIBRE	16
2.3	MANUAL DE USUARIO	18
2.3.1	<i>Herramientas de edición de texto</i>	19
2.3.2	<i>Herramientas de edición de gráficos y video</i>	23
2.3.3	<i>Herramientas colaborativas y de gestión de contenidos</i>	25
2.3.4	<i>Herramientas de documentación de software</i>	31
3	PROCEDIMIENTO PROPUESTO	35
3.1	DESCRIPCIÓN DEL PROCEDIMIENTO	35
3.1.1	<i>Selección, instalación y Configuración de herramientas</i>	36
3.1.2	<i>Estructuración del manual</i>	37
3.1.3	<i>Gestión de roles</i>	39
3.1.4	<i>Edición de contenidos</i>	41
3.2	VERIFICACIÓN DEL PROCEDIMIENTO	42

3.2.1	<i>Selección e instalación</i>	43
3.2.2	<i>Estructuración del manual</i>	44
3.2.3	<i>Gestión de roles</i>	46
3.2.4	<i>Edición de contenidos</i>	47
4	CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	49
5	REFERENCIAS	51
6	ANEXOS	52

Listado de Imágenes

Imagen 1 estándar IEEE 1074, grupos de procesos	8
Imagen 2 Estándar 12207 (grupos de procesos).....	12
Imagen 3 las 4 libertades del software libre	16
Imagen 4 Framapad.....	20
Imagen 5 MarkdownPad (http://markdownpad.com/)	22
Imagen 6. GIMP	23
Imagen 7 tomando captura de pantalla con Apowersoft	24
Imagen 8. WordPress.	26
Imagen 9 Dotclear	28
Imagen 10. Wikis	29
Imagen 11 Proceso Creación Manual de Usuario	35
Imagen 12 Edición en página principal por Usuario2.....	47
Imagen 13 página de moderación de contenidos en la wiki.....	48

Listado de tablas

Tabla 1 Actividades de documentación según estándar 1074.....	11
Tabla 2 gestión de roles.....	40
Tabla 3 Asignación de roles	46

1 INTRODUCCIÓN

La documentación de usuarios es de vital importancia en cualquier tipo de software, y típicamente estos procesos han sido considerados como secundarios en distintas organizaciones que desarrollan software pero no es su razón social el desarrollo de productos software, como por ejemplo la Universidad de Pamplona en la cual se tiene una dependencia que desarrolla software de diferente índole como soporte a sus procesos académicos y administrativos.

El trabajo de la creación de manuales de usuario se le asigna a una o dos personas y se considera tarea que por lo general se realiza al finalizar el desarrollo de la aplicación software, una forma alternativa para hacer este proceso sería la creación de una comunidad colaborativa de fácil participación para cooperar en la creación adecuada de buenos manuales de usuario de acuerdo a las necesidades de cada versión o producto de software y que este se puede ir creando en paralelo al desarrollo de cada funcionalidad del aplicativo.

En este trabajo se definió como estrategia para elaboración de manuales el uso de herramientas de trabajo cooperativo como las wikis, teniendo en cuenta características como la administración de usuarios y la gestión de los contenidos de la herramienta. La estrategia se validó aplicándola en un proyecto simulado de desarrollo de software, al cual se le hizo un manual de usuario de acuerdo al procedimiento planteado en este trabajo.

Este trabajo está basado en experiencias obtenidas por diferentes autores que han participado en comunidades de código abierto (Open Source) y también propias en la participación de conversión ajustes y aplicación de proceso de ayudas en línea del módulo de

facturación y cartera en Gestasoft contratos 4.0 de la Universidad de Pamplona, se espera que este procedimiento basado en modelos de colaboración en comunidades de software libre, pueda a futuro ser adoptado por la Universidad y mejorar sus procesos de desarrollo.

Este documento está organizado en seis capítulos que contienen la estructura en el siguiente orden Introducción, Marco teórico, Procedimiento propuesto, Conclusiones, recomendaciones y trabajo futuro, Referencias y Anexos.

El primer capítulo describe el enfoque del proyecto tanto como los objetivos, el por qué y para que se realiza y una idea de cómo implementar. El segundo capítulo describe todos los conceptos necesarios para poder realizar el proyecto. El tercer capítulo muestra paso a paso el procedimiento propuesto y su respectiva verificación. El cuarto capítulo habla de lo aprendido y lo que se espera a futuro. El quinto capítulo presenta los estudios que se referencian el documento. Y finalmente el último capítulo adjunta el proceso de instalación de la herramienta usada en el procedimiento.

1.1 PLANTEAMIENTO DEL PROBLEMA

En la Universidad de Pamplona una dependencia se dedica al desarrollo de software, entre ellos módulos académicos como Academusoft que es el software que se usa para gestionar todo lo relacionado con los procesos de soporte a los estudiantes desde su inscripción hasta su graduación, en la Universidad y en otras 46 instituciones de educación superior más. Otro tipo de herramienta para los procesos administrativos como Gestasoft para todo lo relacionado con las finanzas y recursos humanos, a pesar de todo esto la Universidad crea e implementa sus propios procedimientos al interior de su organización y no son de disposición para personas externas a ese proceso.

Es por esto que una alternativa a los procesos de desarrollo privados que existen es crear y hacer partícipe de algunos procesos como la elaboración de manuales de usuario mediante una comunidad libre propia de la Universidad, en la que partiendo de todos los trabajos ya realizados por estudiantes de ingeniería de sistemas se pueda fortalecer los procesos de desarrollo en la comunidad.

Debido a la necesidad de generar una guía para el proceso de desarrollo de software en la Universidad de Pamplona, se hace evidente generar una estrategia en la que toda la comunidad universitaria pueda participar y que permita atenuar la falta de conocimiento y participación de esta en la creación de aplicaciones software, por esta razón es que puntualmente en este trabajo se plantea un procedimiento para la creación de documentación de software más específicamente en la creación de manuales de usuario para aplicaciones web.

1.2 OBJETIVOS

1.2.1 Objetivo general

- Diseñar una estrategia para la elaboración de manuales de usuario en ambientes colaborativos y distribuidos.

1.2.2 Objetivos específicos

- Estudiar el funcionamiento de comunidades *open source*, principalmente en el proceso de elaboración de manuales de usuario.
- Definir un proceso para la creación de manuales de usuario, mediante una estrategia de fácil aplicación en el contexto de la Universidad de Pamplona.

- Validar la estrategia selecciona mediante la elaboración de prototipo de manual de usuario de una aplicación ya desarrollada.

1.3 JUSTIFICACIÓN

Los proyectos de código abierto han sido la inspiración de muchas personas y han dado paso a la innovación y mejora de muchos de los productos software, cosa que no permiten los software privados, por ejemplo en el caso de mejorar características o corregir fallos hay alguien siempre con el conocimiento y la disposición de trabajar en algo que va a ser de utilidad para un gran número de usuarios, que no siempre pueden comprar licencias de todos los productos software que existen en el mercado.

En el desarrollo de software como en la invención de muchos otros productos, siempre hay alguien que ya ha realizado una parte del proceso o características completas que podrían servir en otras creaciones, pero por lo general siempre hay que volver a crear desde cero, esto se podría evitar si se pudiera mejorar o innovar en vez de volver a crear lo que ya existe y se podría alcanzar una mejor calidad de software cada vez.

En las comunidades de software libre como en toda organización los productos que se desarrollan necesitan ser usados y administrados por alguien, sin un adecuado manual de usuario no se aprovecharía al máximo un producto o se echaría a perder. En las comunidades de software libre no existe una metodología estándar establecida para documentar software, por lo tanto cada una lo hace de manera diferente de acuerdo a sus necesidades. Con el desarrollo de una metodología para realizar manuales de usuario se busca tener una guía en el proceso de creación de nuevas aplicaciones web, lo que permite tener un camino a seguir para

los nuevos proyectos. Esta metodología se enfoca en los ambiente colaborativos, siendo esta la principal ventaja de aplicar herramientas de gestión de contenidos como camino en la creación de manuales de usuario.

1.4 METODOLOGÍA

En este trabajo se implementó una estrategia para la creación de manuales de usuario, teniendo en cuenta que en la Universidad de Pamplona es necesario establecer un proceso en el desarrollo inspirado en las comunidades de software libre y buscar que se fomente el trabajo en grupo o más bien conformar una comunidad libre que sea motivada a medida de que cada interés individual puede ser tenido en cuenta y generar crecimiento en un corto y largo plazo.

Se recopiló información sobre herramientas que se pueden usar para la elaboración de manuales de usuario entre ellas editores de texto, capturadores de pantalla y principalmente herramientas de edición colaborativa.

Luego a partir de esa información obtenida se plantea un procedimiento que busca crear manuales de usuario con herramientas colaborativas, donde se permita crear contenidos y editarlos por un grupo de personas asignadas a esas tareas para finalmente ser publicado y leído por un usuario final.

Por último se verifica ese procedimiento con un proyecto piloto al que se le hará un prototipo de manual de usuario usando la herramienta seleccionada, haciendo el ejercicio con en un proyecto real, creando usuarios, asignando roles y añadiendo contenidos en una herramienta seleccionada para la gestión del Manual de Usuario.

2 MARCO TEÓRICO

En este capítulo se sintetiza los conceptos más importantes para la definición de la estrategia de creación de manuales de usuario, donde se puede encontrar la información pertinente en la que se basa este proyecto, también algunos autores que hablan del trabajo colaborativo y documentación para usuarios.

2.1 INGENIERÍA DEL SOFTWARE

Según la IEEE La ingeniería de software es: *La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software.*

La Ingeniería del software es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelve problemas de todo tipo.

Permite construir softwares complejos, es importante porque afecta a casi todos los aspectos de la vida y ha invadido el comercio, la cultura y las actividades cotidianas, la ingeniería del software permite crear software en un tiempo razonable y con alta calidad.

Una gran cantidad de organizaciones de desarrollo de software implementan metodologías para el proceso de creación de sus aplicaciones, para toda clase de procesos existen estándares internacionales entre ellos la IEEE 1074 y el ISO 12207 que describe el método de selección, implementación y monitoreo del ciclo de vida del software, en los cuales se hace evidente la necesidad de documentar el software incluyendo la elaboración de manuales de usuario.

2.1.1 Estándar 1074

A continuación se describe las fases o subprocesos que conforman el proceso base de construcción de software correspondiente al estándar IEEE 1074-1989.

El Estándar determina el conjunto de actividades esenciales, no ordenadas en el tiempo, que deben ser incorporadas dentro de un modelo de ciclo de vida de un producto software. Este modelo es seleccionado y establecido por el usuario para el proyecto a desarrollar, ya que la norma no define un ciclo de vida particular.

El proceso base para la construcción de software consiste en analizar las necesidades de la organización en un dominio, bajo un marco de gestión, seguimiento, control y gestión de la calidad. El proceso de software está compuesto de cuatro grupos de procesos principales cada uno de los cuales agrupa una serie de actividades que se encargan de la realización de sus requisitos asociados. Estos son los siguientes:

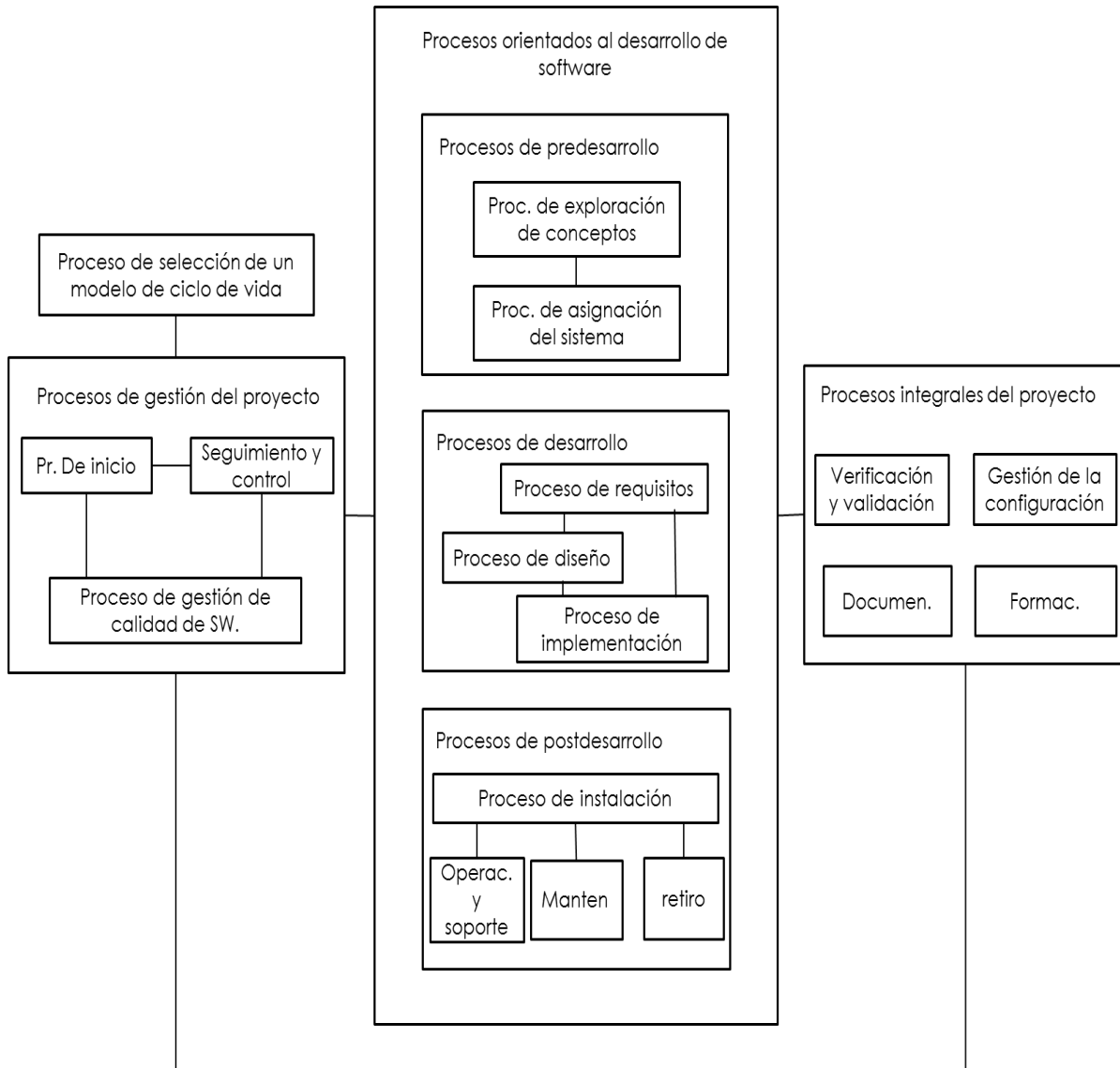


Imagen 1 estándar IEEE 1074, grupos de procesos

2.1.1.1 Procesos de gestión del proyecto

La gestión del proyecto presupone establecer condiciones para el desarrollo del mismo, la gestión involucra actividades dentro de las cuales se tiene:

La planificación de proyectos, define la predicción de la duración de las actividades y tareas a nivel individual, los recursos requeridos, la concurrencia y la superposición de tareas para que sean desarrollados en paralelo y el camino crítico a través de la red de actividades.

La estimación, que se define como la predicción de personal, el esfuerzo y costos que se requerirá para terminar todas las actividades y productos conocidos asociados con el proyecto.

La determinación del tamaño del producto a desarrollar, que es una de las primeras tareas en la gestión del proyecto, ya que sin conocerlo adecuadamente, no es posible planificar y estimar el esfuerzo necesario. El tamaño se define como la cantidad de código fuente, especificaciones, casos de prueba, documentación del usuario y otros productos tangibles que son la salida del proyecto.

2.1.1.2 Procesos de pre-desarrollo

Son los procesos que se deben realizar antes de que comience el desarrollo propiamente dicho del software. El desarrollo se inicia con la identificación de una necesidad de automatización. Esta necesidad para ser satisfecha necesita de una nueva aplicación, o cambio de todo o parte de la aplicación existente. El proceso de pre-desarrollo abarca desde el reconocimiento del problema hasta la determinación de los requisitos funcionales a nivel de sistema, pasando por el estudio de viabilidad de la solución software.

2.1.1.3 Procesos de desarrollo

Son los procesos que se deben realizar para la construcción del producto software. Estos definirán qué información obtener y como estructurar los datos, qué algoritmos usar para procesar los datos y cómo implementarlos, y qué interfaces desarrollar para operar con el

software y cómo hacerlo. A partir del informe de la necesidad, con el soporte de las actividades de los procesos integrales y bajo el plan de gestión del proyecto, los procesos de desarrollo producen el software (código y documentación).

2.1.1.4 Procesos de post-desarrollo

Son los procesos que se deben realizar para instalar, operar, soportar, mantener y retirar un producto software. Una vez terminada la prueba del software, éste está casi preparado para ser entregado a los usuarios finales. Sin embargo, antes de la entrega se llevan a cabo una serie de actividades de garantía de calidad para asegurar que se hayan generado y catalogado los registros, y documentos internos adecuados, que se ha desarrollado una documentación de alta calidad para el usuario, y que se han establecido los mecanismos apropiados de control de configuraciones.

Tan pronto como se entregue el software a los usuarios finales, el trabajo del ingeniero del software cambia, en este momento el enfoque pasa de la construcción al mantenimiento; corrección de errores, adaptación al entorno y mejora de la funcionalidad. En todos los casos, la modificación del software no solo afecta al código, sino también a la configuración entera, es decir, a todos los documentos, datos y programas desarrollados en la fase de planificación y desarrollo.

2.1.1.5 Procesos integrales del proyecto

Son procesos simultáneos y complementarios a los procesos orientados hacia el desarrollo. Incluyen actividades imprescindibles para que el software construido sea fiable (procesos de verificación y validación, gestión de la configuración) y sea utilizado al máximo de sus capacidades (procesos de formación, documentación). Los procesos integrales

comprenden dos tipos de actividades: Aquellas que se realizan discretamente y se aplican dentro de un ciclo de vida del software y las que se realizan para completar otra actividad, estas solo se invocan y no se aplican dentro del ciclo de vida para cada instancia.

Proceso de desarrollo de documentación

El proceso de desarrollo de documentación para el desarrollo y uso del software es el conjunto de actividades que planifican, diseñan, implementan, editan, producen, distribuyen y mantienen los documentos necesarios para los desarrolladores y los usuarios. En la siguiente tabla se identifican las actividades a realizar, y la documentación.

Tabla 1 Actividades de documentación según estándar 1074

Actividades a realizar	Documentación de salida
<ul style="list-style-type: none"> - Planificar la documentación. - Implementar la documentación. - Producir y distribuir la documentación. 	<ul style="list-style-type: none"> - Plan de la documentación.

2.1.2 Estándar 12207

El estándar establece una arquitectura de alto nivel del ciclo de vida del software. Este comienza con una idea o una necesidad que puede ser satisfecha en su totalidad o en parte por el software y termina con el retiro de éste. La arquitectura se construye con un conjunto de procesos e interrelaciones entre estos. La derivación de los procesos se basa en dos principios básicos: la modularidad y la responsabilidad.

Los procesos se agrupan en tres grandes clases: primarias, soporte y de organización.



Imagen 2 Estándar 12207 (grupos de procesos)

Cada proceso está diseñado en términos de sus propias actividades, cada una de las cuales está diseñado además en términos de las tareas que las componen.

Este estándar contiene un conjunto de ocho procesos de apoyo. Un proceso de apoyo es compatible con cualquier otro proceso como una parte integral con un propósito distinto que contribuye al éxito y la calidad del proyecto. Un proceso de apoyo se invoca, según sea necesario, mediante la adquisición, suministro, desarrollo, operación o proceso de mantenimiento, o en otro proceso de apoyo.

Es cierto que existen diversas metodologías y formas de desarrollar software, la realidad es que hay modelos tan antiguos que ya son como la esencia al momento de ver un ciclo de vida de un software. Un ejemplo de esto, es el modelo en cascada para el proceso de desarrollo de un software, en el cuál se ve a ciencia cierta el proceso básico, del cual muchos modelos más suelen surgir.

El Proceso Básico del Ciclo de Vida de un Software tiene varias etapas como son planificación, implementación, pruebas, documentación, despliegue y mantenimiento.

Muchas metodologías de lo que es el ciclo de vida de software, van creando documentación, conforme se va avanzando en el desarrollo del software. Sin embargo algunas otras prefieren no hacer la documentación hasta el final. Sea cual sea la metodología que se elija, la documentación siempre será importante, pues considera que no siempre va a estar una misma persona y su equipo disponibles, y cuando otro equipo llegue a programar lo que hicieron, será indispensable que haya una documentación de la cual se puedan basar, para poder empezar a desarrollar nuevamente el software incompleto.

2.1.3 Proceso de documentación

Dentro de diferentes metodologías estándares y modelos de procesos, la documentación es considerada un proceso para registrar la información producida durante el ciclo de vida. El proceso define las actividades para planificar, diseñar, desarrollar, editar, distribuir y mantener los documentos necesarios por todos los interesados, tales como gerentes, ingenieros y usuarios del software.

Cuando se habla de documentación en general se puede hacer referencia a diferentes tipos de documentación como son:

Documentación de arquitecturas de software

Cuando se utiliza la documentación como base para la comunicación, es necesario para un nuevo lector determinar donde se puede encontrar información particular(Clements et al., 2003).

La documentación de una arquitectura puede ser usada para distintos propósitos a corto, mediano o largo plazo, los cuales se pueden agrupar en tres propósitos globales de acuerdo a algunos autores(Bachmann et al., 2001).

Documentación para el usuario

La documentación para el usuario constituye un elemento de consulta para toda aquella persona que va a usar el programa por primera vez o que trata de saber si el programa servirá a sus objetivos. Igualmente es útil para usuarios que ya realizan un manejo básico y quieren profundizar hacia un conocimiento avanzado.

Más allá de todo esto, se debe tener en cuenta que la estadística demuestra que los usuarios no leen los manuales a menos que no les quede otra opción. Las razones pueden ser varias, pero un análisis de la realidad muestra que se recurre a los manuales solamente cuando se produce un error o se desconoce cómo lograr algo muy puntual, y recién cuando la ayuda en línea no satisface las necesidades del usuario. Por lo tanto, si bien es cierto que se debe realizar manuales, la existencia de un buen manual nunca libera de hacer un producto amigable para el usuario, que incluso contenga ayuda en línea. Es incluso deseable proveer un software tutorial que guíe al usuario en el uso del aplicativo, con apoyo multimedia, y que puede llegar a ser un curso online(Suárez & Fontela, 2003).

La extensión de la documentación para el usuario será variable en función de la complejidad y características del programa: puede ir desde un párrafo para programas muy sencillos y de fácil uso hasta centenares de páginas para programas comerciales complejos. Los puntos contenidos en la documentación también son variables. Para programas sencillos puede reducirse a un título, una explicación breve del funcionamiento, entradas y salidas y un ejemplo de uso.

Documentación para mantenimiento

La documentación para mantenimiento constituye el elemento de referencia para el programador que haya de realizar cambios o ampliaciones del programa en el futuro. La necesidad de mantenimiento deriva de:

- Defectos del programa no detectados y que es necesario corregir.
- Cambios externos de índole política, técnica, social, etc. que afectan al programa: normativa, moneda, novedades de un sistema operativo, etc.
- Solicitudes de los clientes o usuarios.

El mantenimiento de un programa puede afectar a su esqueleto o diseño básico, a funciones importantes pero desligadas del núcleo del programa o a cuestiones meramente estéticas. De cualquier forma, el mantenimiento debe considerarse como programación en todos sus sentidos, debiendo partir del conocimiento del problema y avanzar con detenimiento siguiendo las normas para una programación sólida. Es ideal un mantenimiento que respete la filosofía y el estilo del programa que se mantiene, de modo que un auditor no pudiera detectar qué parte del programa corresponde al código original y qué parte a la ampliación o corrección.

2.2 OPEN SOURCE Y SOFTWARE LIBRE

A pesar de que son términos relacionados y se suelen usar indistintamente, Software Libre y Open Source no son exactamente lo mismo, al menos no según la FSF¹, organización creada por Richard Stallman en el año 1985 con el propósito de difundir este movimiento.

Software libre

Según Richard Stallman que es como el padre del software libre, - el **Software Libre** es un asunto de libertad, no de precio. Para entender el concepto, debes pensar en "free"² como en "libertad de expresión", no como en "cerveza gratis". Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

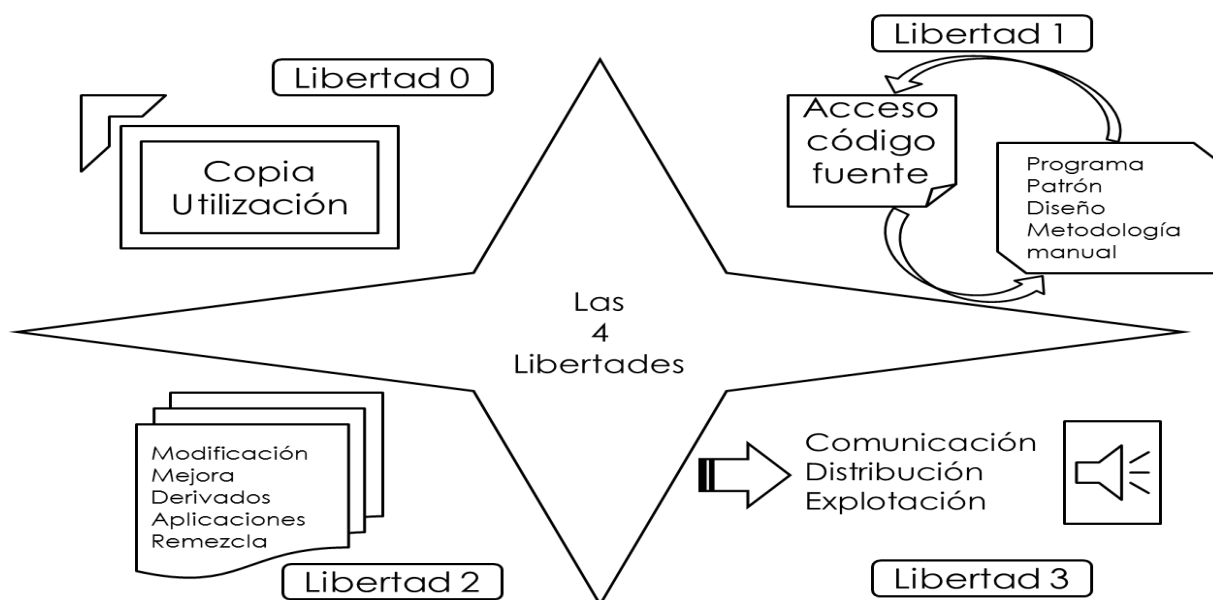


Imagen 3 las 4 libertades del software libre

¹ iniciales de Free Software Foundation

² en inglés una misma palabra (free) significa tanto libre como gratis, lo que ha dado lugar a cierta confusión.

-La libertad de usar el programa, con cualquier propósito (libertad 0).

-La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.

-La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).

-La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (Libertad 3). El acceso al código fuente es un requisito previo para esto.

Software libre no significa no comercial. Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual.

Open Source

Se puede apreciar que el origen del Open Source está muy relacionada con el software libre, de hecho para poder poner en práctica las libertades (1)³ y (3)⁴ se debe tener acceso al código fuente del producto software, es así que el proyecto GNU expone que para el año de 1998 un sector de la comunidad de software libre decidió tomar un nuevo rumbo para promover el Open Source.

Dentro de este nuevo movimiento cabe destacar la participación de Erick Raymond y Bruce Perens, este grupo de personas ve los beneficios para el software que se producen cuando el código está disponible para todos, luego el concepto filosófico de Open Source se basa en la idea de mejorar el software en un sentido esencialmente práctico, dejando de un

³ realizar cambios

⁴ publicar versiones modificadas

lado las ideas filosóficas de las libertades del software libre, lo cual refleja que este nuevo movimiento deja a un lado la esencia de lo que es software libre.

Diferencias entre Software Libre y Open Source

Como se puede ver, el movimiento del Software Libre hace hincapié en cuestiones éticas y morales relacionadas con el software, viendo el aspecto técnico como secundario, a diferencia del movimiento Open Source que lo establece como prioritario, siendo esta su diferencia más grande.

Otra de sus diferencias, quizá la más importante para lo que al ámbito comercial respecta, es que con una aplicación desarrollada bajo los estándares del Software Libre puedes obtener remuneración por conceptos de desarrollo, soporte y puesta a punto siempre y cuando se entreguen los códigos fuente, a diferencia del movimiento Open Source que no obliga a hacerlo. Todos los productos desarrollados en Software Libre así como sus derivados siempre deben ser libres, a diferencia de Open Source.

Básicamente el movimiento de Software Libre tiene diferencias de índole filosóficas con el Open Source, sin embargo, ambos han permitido el desarrollo de software de gran calidad y con muchísimas más libertades de las que el software privativo otorgaría jamás.

2.3 MANUAL DE USUARIO

En el desarrollo de software, el proceso de documentación está ligado a documentar requerimientos, código, arquitectura, guías de usuario, ayudas en línea entre otros. Sin embargo este trabajo está enfocado únicamente en la documentación para usuarios finales, es decir manuales o guías de usuario.

Un manual de usuario es, un documento de comunicación técnica que busca brindar asistencia a los sujetos que usan un software. Más allá de su especificidad, los autores de los manuales intentan apelar a un lenguaje ameno y simple para llegar a la mayor cantidad posible de receptores.

Dada su complejidad, todos los productos electrónicos o informáticos suelen contar con su propio manual de usuario. Los artículos más simples (como una pelota o una mesa) no requieren de explicaciones para que los consumidores sepan cómo utilizarlos.

Los manuales de usuarios suelen estar escritos en diversos idiomas y contar tanto con textos como con imágenes. De esta forma se facilita la comprensión de los conceptos. Los diagramas y esquemas también son habituales en cada uno de ellos.

La mayor deficiencia de los sistemas operativos no reside en el software, sino en la ausencia de buenos manuales libres para los softwares. La documentación es una parte esencial de cualquier producto software, un paquete importante de software libre sin un buen manual libre que lo acompañe constituye un lastre considerable(Stallman, 2002).

Algunas herramientas sugeridas para la construcción de estos manuales de usuario se describen a continuación.

2.3.1 Herramientas de edición de texto

Trabajar en un texto que tiene que ser validado por mucha gente, no es divertido después de cinco minutos cuando todo el mundo tiene su opinión y que para que cada uno pueda darla pasan días/semanas y tal vez un montón de correos.

En el mundo del open source y que no muchos conocen existe el trabajo colaborativo que esta habitualmente ocupado por programas y/o aplicaciones de tipo freemium⁵ que buscan cobrar más por ciertas "características avanzadas".

Es por ello que se han inspirado proyectos que han simplificado las cosas al extremo y ni hay que instalar nada en el PC, todo se sucede directamente en la web, ni hay que registrarse para empezar a crear un pad⁶.

Algunas herramientas sugeridas serían, Framapad, TitanPad y MeetingWords. El usuario solo tiene que crear un pad, cada pad tiene una URL única a compartir con sus colaboradores, se puede determinar si el pad estará disponible durante un día, una semana, un mes o si es ilimitado después de la última entrada escrita. Cada colaborador puede elegir un color de fondo de su texto para ser identificado fácilmente y un chat integrado permite a los participantes charlar entre si directamente.

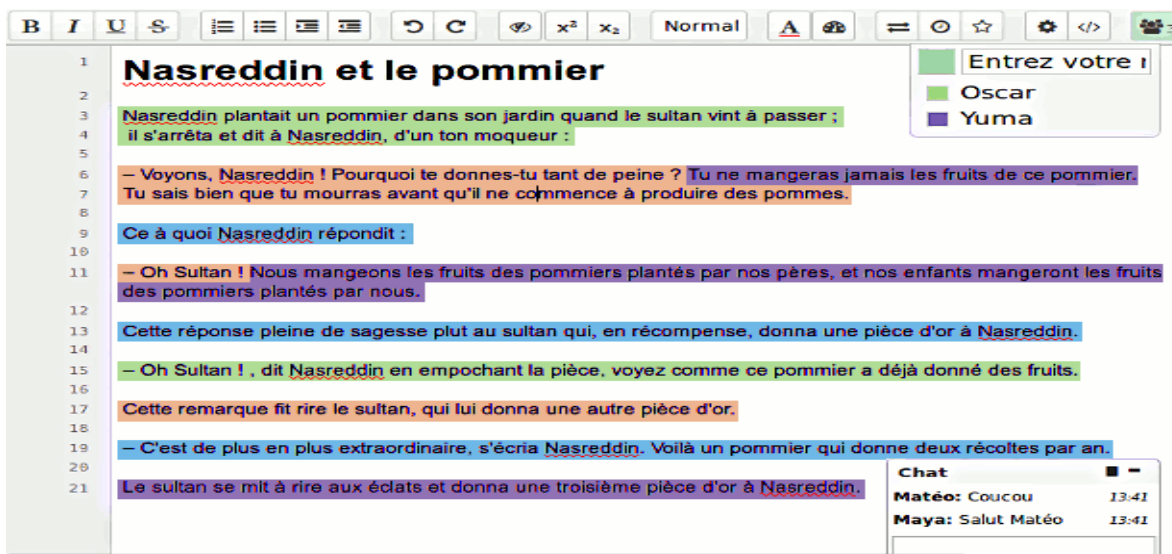


Imagen 4 Framapad

⁵ freemium es una contracción en inglés de las dos palabras: "free" y "premium".(gratis y de pago)

⁶ Pad es un editor de texto colaborativo en línea.

Los pads ofrecen todas las funciones necesarias para la importación y exportación de un documento en formato HTML, Word, texto, PDF, puede resultar práctico si se necesita trabajar sobre un texto existente y guardarlo en caso de problemas eventuales del servidor o para trabajar sin conexión a internet.

Algunas herramientas de este tipo son:

FramaPad → <https://framapad.org/>

TitanPad → <https://titanpad.com/>

(Tutorial herramienta - <https://www.youtube.com/watch?v=fxyzLmq7Zio>)

MeetingWords → <http://meetingwords.com/>

MarkdownPad

Con una versión gratuita y Premium, MarkdownPad es un editor popular para Windows. Está optimizado para publicaciones en blogs, sitios web, artículos, LEEMEs y documentación de software.

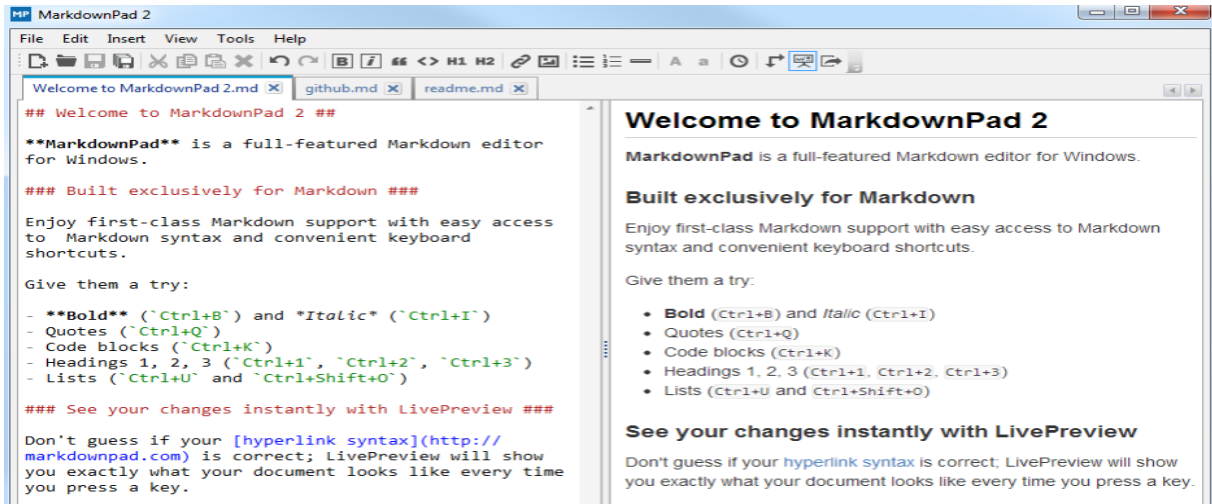


Imagen 5 MarkdownPad (<http://markdownpad.com/>)

Editor reStructuredText

Markdown es uno de los más utilizados para escribir la documentación del software, pero hay otro que no se ha descrito hasta ahora, y que es reStructuredText. Es muy similar a Markdown, pero vale la pena aprender para fines de documentación de software.

Docutils, el creador de reStructuredText, ha creado una lista de editores reStructuredText, que incluye:

- Un complemento para vim.
- Emacs (en modo rst).
- Un complemento para Eclipse.
- Un complemento para TextWrangler / BBEdit.
- NoTex (para navegadores).

La idea de reStructuredText es que es fácil de convertir entre diferentes formatos, especialmente de texto sin formato a un sitio web estático.

2.3.2 Herramientas de edición de gráficos y video

Las imágenes son una parte fundamental cuando se realiza documentación de software especialmente cuando se hace un manual de usuario, por tanto es de importancia prestar atención a una buena calidad de imagen, explícita e intuitiva. Algunas herramientas alternativas de código abierto son:



GIMP como herramienta popular de código abierto para edición de imágenes, tiene una función de capturar pantalla bastante útil. Se puede instalar en Windows y en los Linux más conocidos ya está por defecto instalada.

(GIMP, <https://www.gimp.org/>)

(Paint.NET, <http://www.getpaint.net/download.html>)

A pesar de las facilidades de las herramientas online para edición de imágenes otra opción de edición, captura y para realizar algún video explicativo según sea el caso de la guía o manual que se esté realizando, se recomienda una última herramienta que aparte de realizar capturas de pantalla permite editarlas al mismo tiempo, también de los mismos creadores existe otra funcionalidad para grabar la pantalla con una resolución personalizada entre otros ajustes. Las guías se usuario respectivas para el uso y descarga de los complementos necesarios para poder usarla adecuadamente están en el siguiente link. (Estas herramientas fueron probadas solamente en el SO Windows).

Capturar pantalla → <https://www.apowersoft.es/captura-de-pantalla-gratis>

Grabador de pantalla → <https://www.apowersoft.es/grabador-de-pantalla-gratis>

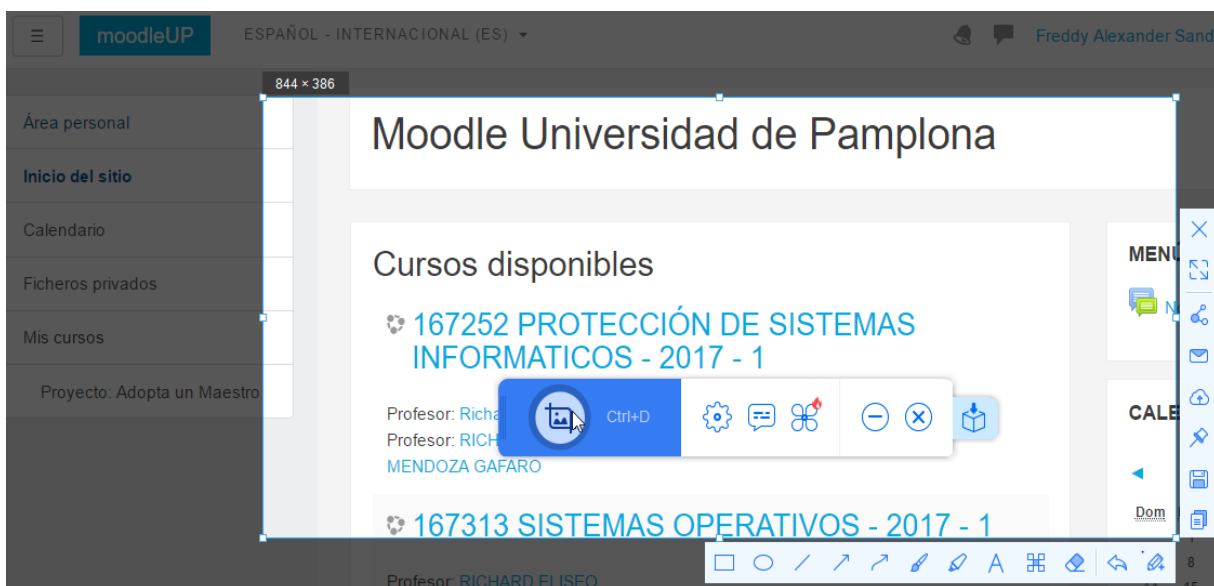


Imagen 7 tomando captura de pantalla con Apowersoft

2.3.3 Herramientas colaborativas y de gestión de contenidos

Las herramientas de trabajo colaborativo tienen como objetivo la construcción de contenidos ya sean libros, documentos y por qué no un manual de usuario en conjunto con muchas personas, los colaboradores no necesariamente tienen que estar en el mismo lugar o al mismo tiempo. Debido a esta flexibilidad para desarrollar proyectos en grupo se hace ideal este tipo de herramientas sin pensar en obligar a las personas, todos realizan tareas por un objetivo en común.

Para que el desarrollo de software y el trabajo en grupo de proyectos colaborativos sean ordenados y acelere las tareas de trabajo, siempre es importante el uso de herramientas para la edición de textos, realización de diagramas y demás procesos que se presentan en el desarrollo de un manual de usuario en software. Se han descrito anteriormente algunas herramientas, de acuerdo a estos procesos existen herramientas que cubren ciertas características para cada necesidad o aspecto que se requiere. En este caso para la publicación y edición colaborativa de los contenidos se verán a continuación unas herramientas sugeridas.

Actualmente existen literalmente miles de herramientas y aplicaciones Web 2.0 disponibles en la Internet. Estas pueden clasificarse en: Publicidad, Blogging, Bookmarks, Catálogos, Chat, Comunidades, Colaborativas, Educativas, Correo, Eventos, News Feeds, Búsqueda, Compras, Etiquetamiento (Tagging), Video, Widgets y Wiki.

Dado que hay literalmente una innumerable cantidad de aplicaciones Web 2.0 disponibles en la Internet, es difícil simplemente escoger a ciegas, alguna de ellas. Por esto mismo, es indispensable establecer algunos aspectos generales en la selección de la herramienta que cubrirá la mayoría de necesidades. Los siguientes son criterios generales para tener en cuenta.

- Sistema multiusuarios y de grupos de usuarios con permisos.
- Que sea de código libre para usar y modificar.
- Que exista buena documentación y guías para usuarios y desarrolladores.
- Control de revisión y aprobación de los contenidos.

2.3.3.1 WordPress

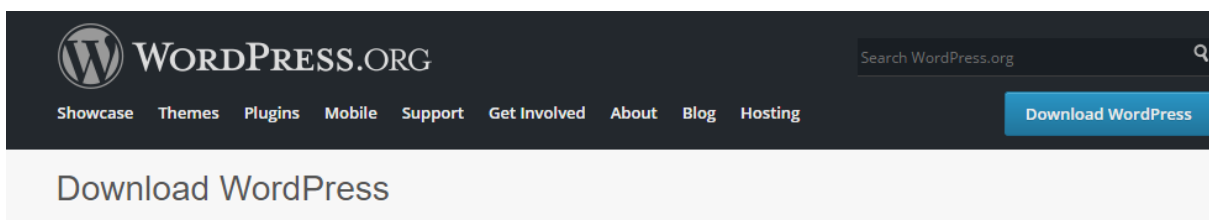


Imagen 8. WordPress.

WordPress posee características para crear páginas que lo convierte en uno de los mejores CMS⁷ que existe actualmente. Es ligero y sencillo de usar y con algunos plugins puede funcionar desde tienda online hasta una red social. Es tener las herramientas a mano y un poco de creatividad para desarrollar infinidad de proyectos con WordPress.

WordPress cuenta con perfiles de usuarios predefinidos, conocidos como roles:

Suscriptor: Este usuario no puede crear contenidos, sólo verlos. Sus capacidades son: ver el panel y editar su propio perfil. Es útil si se crea una web privada en la que la web sólo la pueden ver los usuarios registrados.

Colaborador: Puede crear borradores de contenido pero no publicarlos ya que el administrador debe aprobarlo primero.

⁷ CMS: Sistema de gestión de contenidos.

Autor: Tiene los privilegios del Colaborador y además puede subir archivos y publicar entradas

Editor: Tiene todos los privilegios del Autor y además puede editar entradas y páginas, moderar comentarios y modificar categorías y enlaces.

Administrador: Todos los privilegios para administrar la web.

Publicación de Contenidos: Puede actuar como gestor de contenidos, como blog o como ambos simultáneamente, lo que le permite disponer de un sitio web empresarial con su blog corporativo, todo ello gestionado de forma sencilla por una única herramienta.

Las páginas y artículos se generan dinámicamente a medida que se publica, por lo que la actualización es fácil y rápida.

Al tener una base internacional, permite tener blogs y sitios web en casi cualquier idioma.

2.3.3.2 *Dotclear*

Dotclear es una herramienta de software Web 2.0 para la publicación de contenidos. El principal objetivo de Dotclear es ofrecer a los usuarios una herramienta amigable para publicar contenidos con facilidad en la Web sin importar el nivel de habilidades técnicas del usuario. Dotclear es gratuito diseñado principalmente para sus usuarios y es regularmente mejorado. Las actividades que se pueden realizar son Blogging, Multi-blogging, Folksonomy, Publicación de Contenidos y Edición.

Dotclear tiene una funcionalidad rica que lo convierte en una herramienta de publicación de alta calidad que puede igualar e incluso superar herramientas similares en algunos aspectos. Dotclear fue desarrollado con PHP y MySQL.

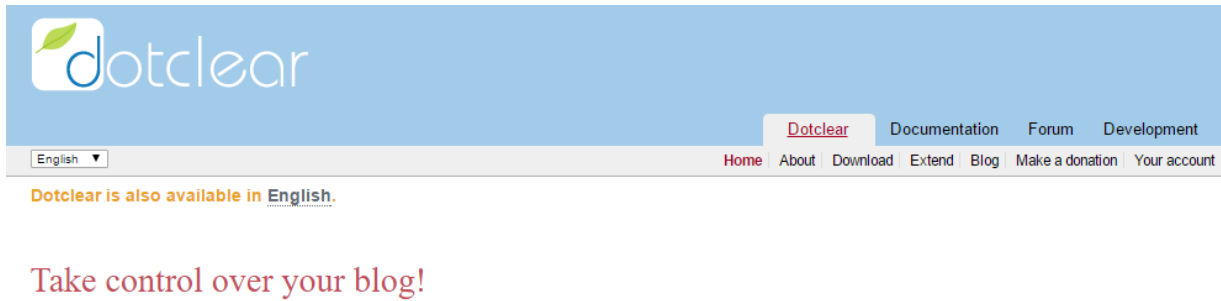


Imagen 9 Dotclear

Principales características.

- Administración de archivos multimedia.
- Edición de sintaxis dual Wiki/XHTML con traductor interno.
- Soporte a múltiples lenguaje.
- Cumple con código W3C.
- Sistema de plantillas flexible.
- Etiquetas y categorías para las páginas.
- Permite entradas flexibles con comentarios y control de cambios.
- Sistema configurable completo de temas de apoyo a los plugin.

2.3.3.3 Wikis

La primera definición de wiki (“What Is Wiki,” 2002) vino propuesta por Cunningham, su desarrollador original, para el cual un wiki es:

«La base de datos en línea más simple que pueda funcionar» (The simplest online database that could possibly work).

Es muy frecuente, por ejemplo, la creación de wikis para proveer de documentación a programas informáticos, especialmente los desarrollados en software libre.

Un wiki es una aplicación informática que reside en un servidor web y a la que se accede con cualquier navegador y que se caracteriza porque permite a los usuarios añadir contenidos y editar los existentes. El nombre wiki se basa en el término hawaiano “wikiwiki” que significa “rápido” o “informal”. La historia de los wikis, tal como se recoge en la Wikipedia, comienza en 1995, cuando el padre de la idea, Ward Cunningham, estaba buscando nombre para una aplicación que había diseñado que permitía que una comunidad dispersa de usuarios editara páginas web rápidamente usando un navegador y formularios web. En su primera visita a Hawai vio que los autobuses que unen entre sí las terminales del aeropuerto se llamaban “wikiwiki”, también fue la primera palabra hawaiana que aprendió, y así se evitó que su aplicación se llamara “quick-web” o algo igualmente aburrido.

Características de las wikis



Imagen 10. Wikis

Una Wiki tiene un lenguaje de marcas muy simple y conciso, que recuerda la presentación final, sin ser exactamente como ella. Por ejemplo, los párrafos se separan por una línea en blanco, los elementos de listas empiezan por un guion si no se numeran y por un cero si se numeran, o las celdas de tablas se separan por barras verticales y horizontales(Wiki - EOI, 2012).

Tampoco existe el concepto de documento completo, sino que un Wiki es más bien un conjunto de pequeños documentos enlazados que se van creando a medida que es necesario explicar un nuevo concepto o tema. La creación de los documentos es casi automática, ya que la herramienta de edición muestra muy claramente que se ha introducido un concepto (a través de un NombreWiki, casi siempre dos palabras juntas con la primera letra en mayúsculas). Casi ningún Wiki admite hiperenlaces dentro de la misma página.

Cualquiera puede escribir en un Wiki, aunque se aconseja que se identifique el autor con un registro previo. Cuando se visita un Wiki se ve que todas las páginas tienen un botón para permitir su edición según está configurada. Si existe, el navegador mostrará en un formulario la fuente del documento, que se puede cambiar. No es una edición WYSIWYG⁸, lo que disuade al que quiera fastidiar, pero es tan sencillo que cualquier interesado puede modificar documentos con muy pequeño esfuerzo.

Los Wikis llevan su propio control de versiones de documentos, de modo que generalmente están accesibles todas sus versiones con indicación de quien las hizo y cuando.

⁸ WYSIWYG : What You See Is What You Get.

También se pueden comparar con facilidad. También suelen llevar mecanismos de búsqueda, al menos por nombre de página y por palabra contenida.

Normalmente el autor original de una página querrá enterarse de las modificaciones que se le hacen. Para ello puede suscribirse a los cambios, recibiendo notificaciones de los mismos por correo electrónico. A veces el que ve un documento no se atreve a cambiar nada, pero puede hacer un comentario. Normalmente toda página wiki tiene asociado un foro de comentarios que se pegan al final del documento y que, bien el autor original, bien alguien que asuma el rol de editor, puede emplearlos para reformar el texto original, posiblemente moviendo frases de los comentarios a los sitios oportunos.

Muchas herramientas ofrecen ventajas a la hora de realizar modelos, diseños o dibujos para los proyectos de software, Wiki no ofrece estas herramientas, sin embargo hay aspectos positivos y negativos a la hora de usar una Wiki versus documentación en Word o herramientas similares. Ver (Bachmann, Merson, Architecture, Initiative, & Cmu, 2005) ahí se define una comparativa con ventajas y desventajas.

2.3.4 Herramientas de documentación de software

Sin documentación, el software es sólo una caja negra. Y las cajas negras no son ni mucho menos tan útiles como podrían ser porque su funcionamiento interno está oculto de los que las necesitan al aire libre.

La documentación del software convierte su software en una caja de vidrio explicando a los usuarios y desarrolladores cómo funciona o se utiliza.

Herramientas para generar automáticamente la documentación desde el código fuente

No hay nada como el toque humano cuando se trata de documentación. Sin embargo, como punto de partida (especialmente para bibliotecas con cantidades de material gigantescas), lo mejor es generar automáticamente la documentación esquelética. Este trabajo analizando las funciones y los comentarios de la fuente, y hay algunas opciones diferentes dependiendo del idioma:

Doxygen (C, C ++, C#, D, Fortran, IDL, Java, Objective-C, Perl, PHP, Python, Tcl y VHDL)

GhostDoc (C #, Visual Basic, C ++ / CLI, JavaScript)

Javadoc (sólo Java)

Docurium (Ruby)

Antes de elegir o depender únicamente de la generación automática, sugeriría consultar información en internet sobre los pros y los contras.

Herramientas usadas en proyectos software libre

Algunas herramientas bastante populares y comúnmente usadas en comunidades libres como GNOME, KDE entre otras hace que deban ser mencionadas teniendo en cuenta su potencial, aunque no sean la herramienta más rápida de usar o de las más documentadas. A continuación se verá una descripción de estas herramientas.

DocBook

DocBook es una aplicación del estándar SGML/XML e incluye una DTD (Document Type Definition) propia y que se utiliza de manera más destacada en el área de la documentación técnica, especialmente para documentar todo tipo de material y programas informáticos. Existe un Comité Técnico de DocBook en OASIS (originalmente SGML Open) (por sus siglas en inglés de Standard Generalized Markup Language) que mantiene y actualiza este estándar. DocBook inicialmente comenzó como una DTD⁹ de SGML¹⁰, pero a partir de la versión 4 existe un equivalente para XML.

Como lenguaje semántico que es, DocBook permite crear documentos en un formato neutro, independiente de la presentación. En este formato neutro se recogen tanto el contenido como la estructura lógica del mismo, permitiendo así que pueda ser publicado (presentado) automáticamente en multitud de formatos: HTML, XHTML, EPUB, PDF, man pages, HTML Help, etc., simplemente aplicando "plantillas" de presentación, sin que sea necesario ningún cambio sobre el documento original.

LinuxDoc

LinuxDoc es un lenguaje similar a DocBook. Este lenguaje es un lenguaje de marcado generalizado estándar o SGML (por sus siglas en inglés de Standard Generalized Markup Language) con estructura DTD definición de tipo de documento (por sus siglas en inglés de Document Type Definition).

Fue creado por Matt Welsh, y su primera versión 1.1 fue liberada en 1994. Este Lenguaje de marcado es utilizado en varios sitios en internet entre los cuales se encuentra la página de Documentación de Linux. Las etiquetas de LinuxDoc son a menudo más cortas que

⁹ DTD -Document Type Definition (definición de tipo de documento).

¹⁰ SGML Standard Generalized Markup Language (lenguaje de marcado generalizado estándar).

las etiquetas de DocBook, además que LinuxDoc tiene más sucinto DTD que DocBook. Esto hace que LinuxDoc tenga mayor preferencia por los usuarios con pequeños a medianos proyectos que han encontrado a LinuxDoc cumplir sus necesidades de una mejor forma que DocBook. Cabe destacar que la distribución Debian de Linux tiene un LinuxDoc-Tools Package integrado.

3 PROCEDIMIENTO PROPUESTO

En el siguiente capítulo se propone una manera o guía para la elaboración de manuales de usuario con herramientas para gestión de contenidos, teniendo en cuenta sus ventajas a la hora de desarrollar colaborativamente es decir gestionar contenidos en conjunto con otras personas. Es importante resaltar que este no pretende ser un tutorial guía a la hora de usar e instalar una herramienta sino que más bien es una estrategia en el desarrollo de proyectos de software libre, que puede ser implementada con cualquier herramienta que tenga características para el manejo de usuarios y la gestión de contenidos colaborativamente.

3.1 DESCRIPCIÓN DEL PROCEDIMIENTO

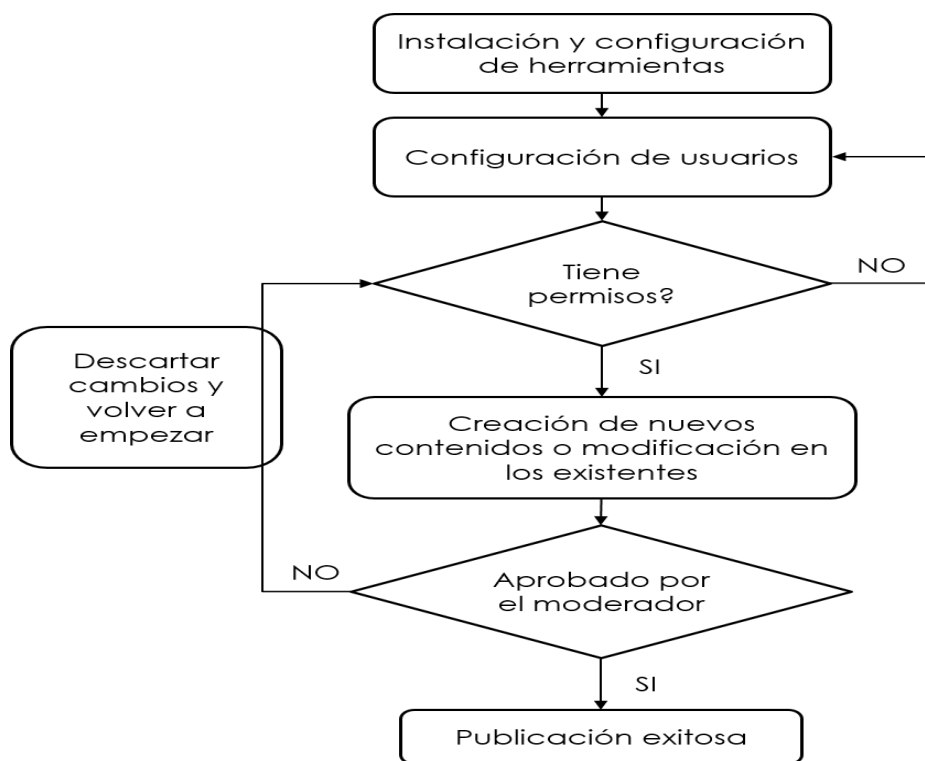


Imagen 11 Proceso Creación Manual de Usuario

3.1.1 Selección, instalación y Configuración de herramientas

En la búsqueda de herramientas con las que se pueda gestionar documentación en específico para realizar manuales de usuario se debe tener en cuenta las necesidades a la hora de poder mostrar la información en algún sitio accesible a cualquier interesado, si permite subir imágenes, editar contenidos con facilidad, control de usuarios y manejo de múltiples proyectos.

Se busca que se permita la gestión de usuarios y un adecuado control, esto significa que los usuarios solo puedan hacer lo que se les permita hacer. Una herramienta que permita conceder permisos individualmente y en grupo a cada usuario, crear grupos de usuarios según sea necesario.

Para una buena gestión del manual se requiere herramientas con las que se pueda hacer una buena gestión del contenido insertando capturas de pantalla o imágenes alusivas de buena calidad. Incluir video tutoriales si fuese necesario, fragmentos de código y cualquier otra ayuda que el usuario pudiese usar en beneficio.

En algunos otros casos es de utilidad poder editar imágenes, hacer capturas de pantalla de calidad o crear una imagen propia (diagrama de flujo, esquema). Por lo que si un gestor de contenidos no trae estas herramientas, se podrían instalar ya sean complementos o usar herramientas online más pertinentes y fáciles de usar.

Se recomiendan herramientas para la gestión y publicación de los contenidos como Wordpress o las wikis por la capacidad para el trabajo colaborativo. La herramienta debe ser

seleccionada de acuerdo a sus intereses o por sus habilidades en el manejo y adaptación de la herramienta a los objetivos que se quieren lograr.

3.1.2 Estructuración del manual

Para realizar un manual de usuario existen muchas guías o esquemas que orientan como se debe llevar, sin embargo siguiendo los estándares de la IEEE se llega a un posible modelo generalizado.

Una estructura sugerida para la creación del manual de usuario con una herramienta para la gestión de contenidos colaborativamente es la siguiente:

Si existe más de un proyecto se sugiere tener una lista que tenga hipervínculos a todos los manuales de usuario existentes, es decir un enlace por cada módulo al que se le creará un manual de usuario.

- * Un prefacio, con información sobre cómo usar el propio manual.

- * Un índice: para que sea más intuitivo, cada manual de usuario de una aplicación se subdivide en todas las funcionalidades que deben ser documentadas, por tanto para cada una se creará una página, en caso de ser cortas se pueden dividir en secciones y así abarcar todo el módulo con facilidad.

- * Una guía rápida sobre cómo usar las funciones principales del software: En caso de que el software tenga funciones generalizadas o especiales se hace oportuno una explicación de los principales botones que aparecerán en todo el aplicativo.

- * Una sección para la resolución de problemas: En toda aplicación siempre existen casos especiales que no siempre están en el flujo normal de los procesos, por tanto se hace

necesario un lugar donde se pueda documentar este tipo de situaciones, también ver las posibles soluciones planteadas por otros usuarios, y en casos graves que esto sirva de soporte a los desarrolladores que solucionan bugs o errores.

* Una FAQ. (Respuestas a preguntas frecuentes): Como en todo lugar siempre hay una especie de reglas iniciales sobre cómo está planteado el software, es necesario enterarse de cómo funciona normalmente y que se puede o no hacer para casos generalizados en la plataforma para la que se crea esta sección.

* Información de contacto.

* Un glosario: Se Relación con las funciones principales del software, pero se describen términos en función de la aplicación software.

De acuerdo al estándar *ISO/IEC, 26514:2008* se recomienda una estructura general que puede ser modificada de acuerdo a lo que se necesite. La documentación según el estándar consta de:

1. Introducción

-Propósito

-Ámbito del Sistema

-Definiciones, Acrónimos y Abreviaturas

-Referencias

-Visión General del Documento

2. Descripción general

- Perspectiva del Producto
- Funciones del Producto
- Características de los Usuarios
- Restricciones
- Suposiciones y Dependencias
- Requisitos Futuros, Interfaces.

3. Requisitos específicos

- Interfaces Externas
- Funciones
- Requisitos de Rendimiento
- Restricciones de Diseño
- Atributos del Sistema
- Otros Requisitos

4. Apéndices

3.1.3 Gestión de roles

Es necesario llevar un control adecuado y ordenado cuando se trata de trabajo colaborativo, repartir tareas y responsabilidades es una buena estrategia para llevar a cabo un objetivo. En el caso de este procedimiento se proponen dos roles que se describen a continuación:

Moderador de Contenidos: El moderador se encarga de revisar y aprobar los contenidos nuevos o modificaciones que suben los demás usuarios de la comunidad. Puede conceder permisos a los demás usuarios o limitar ciertas funciones de edición. Hay que tener en cuenta que si un cambio no es significativo aun así debe ser revisado y aprobado por el respectivo moderador.

Editor: Es un usuario estándar, en este caso aquel que es miembro importante por su trabajo en la comunidad pero no tiene permisos para borrar páginas o asignar permisos a otros usuarios por encima de los que posee actualmente. Este se encarga de la creación de nuevos contenidos o modificaciones en los existentes, podrá introducir texto e imágenes, al momento de subir los nuevos datos son revisados y aceptados por los moderadores.

Una vez conseguida la distribución de los respectivos roles se organizan con los módulos en que va a trabajar cada editor y moderador en el manual de usuario a través de una tabla.

Tabla 2 gestión de roles

Roles Usuarios	Moderador	Editor
Usuario 1	x	
Usuario 2		x

3.1.4 Edición de contenidos

La edición electrónica de todo el material que se va a mostrar al usuario final es mayormente contenido multimedia que puede combinar desde el texto hasta video tutoriales guiados.

El texto

El texto generado por los editores ya sea generado de su propia autoría o transcrito de otros manuales, será introducido en la herramienta y transformado en un archivo informático. Como es un ambiente hipertextual la lectura se suele realizar de forma no lineal diferente a como se hace en los libros impresos. Los usuarios no están obligados a seguir una secuencia establecida, sino que pueden moverse a través de la información y hojear intuitivamente los contenidos, siguiendo sus intereses en búsqueda de un término o concepto.

Imágenes

Para el tratamiento de imágenes digitalizadas son varias las herramientas de software que se pueden utilizar. Unas aplicaciones permiten buscar, capturar, obtener, ordenar y transferir imágenes, facilitando el manejo de grandes cantidades de información multimedia. Otras aplicaciones ayudan a manipularlas, a modificar sus parámetros fundamentales, a eliminar o añadir elementos, a combinarlas con otras imágenes o a aplicarles filtros hasta obtener los resultados deseados.

En cuanto a los formatos que se pueden emplear tres son los principales:

GIF (Graphic Interchange Format) Es utilizado tanto para ilustraciones como también animaciones.

JPEJ (Joint Photographs Experts group). Es el más adecuado para tratamiento de fotografías.

PNG (Portable Network Graphic). Es un formato pensado para ser distribuido directamente en la red, y el más reciente de los tres.

El vídeo digital

Es bastante útil aunque el problema de la digitalización de imágenes en movimiento es que necesitan de una elevada compresión. El principal propósito en de la compresión en el tratamiento digital del audio y del vídeo es conseguir que el ancho de banda promedio y máximo se mantenga dentro de unos límites aceptables. El punto clave de la compresión es la compilación gráfica que determina todos los parámetros y opciones para un trabajo dado.

El estándar más extendido es MPEG (Motion Pictures Experts Group), con una relación de compresión de 1/200. Existen otros softwares como Vídeo for Windows (de Microsoft) y QuickTime (de Apple). Así para el tamaño de la imagen los mejores resultados se obtienen con tamaños pequeños, del orden de 160×120 ó 320×240 pixels como máximo.

En cuanto a la velocidad de reproducción el comportamiento ideal se halla entre los 15 y los 25 fotogramas por segundo, para un ordenador de gama media.

3.2 VERIFICACIÓN DEL PROCEDIMIENTO

Para verificar la estrategia definida se eligió un manual ya existente y se selecciona una herramienta para realizar el proceso de creación de este manual de usuario, luego se describe la estructura del manual y se asignan los roles correspondientes por manual, por último algunas anotaciones sobre la edición de los contenidos.

3.2.1 Selección e instalación

Para la construcción de esta propuesta se seleccionó *MediaWiki* como herramienta de edición y publicación de los contenidos en la creación del manual de usuario, debido a sus funciones para la configuración de usuarios y fácil lenguaje en la edición de contenidos. Este software se ha hecho bastante popular en proyectos colaborativos como lo es Wikipedia para lo que fue orientado inicialmente, actualmente es utilizado también por varios proyectos más de la Fundación *Wikimedia* sin ánimo de lucro y por muchos otros wikis, incluyendo el sitio web, hogar de *MediaWiki*.

Como primer paso, se instaló un servidor LAMP usando las herramientas nombradas a continuación en un servidor del programa de ingeniería de sistemas del proyecto de software libre (MediaWiki, 2016).

- Linux (Debian en este caso)
- Apache2
- Mysql
- PHP5

En la instalación de *MediaWiki* se debe tener información esencial como:

Nombre de la base de datos - [upwiki]

Contraseña del Mysql [****]

Creación de usuario principal y contraseña - [freddysandoval] – [*****] – Con este usuario se configuraran los demás usuarios de la wiki asignando permisos gráficamente.

Al usar la Wiki para crear un Manual de usuario de una aplicación, se sugieren unas recomendaciones generales que son oportunas a la hora de configurar por primera vez la Wiki como también cuando se comienza a desarrollar el contenido.

- Restringir la edición de las páginas a todo el mundo, esto conlleva a la creación de usuarios y asignación de permisos ver (3.1.3).
- Decidir qué aspecto llevara la Wiki, aunque no es relevante en aspecto técnico si influye como la verán los usuarios finales, también de ello puede depender el uso y la precepción de la información. Para ello existe una web donde hay muchos estilos de apariencia(“skins,” 2016). O simplemente usar los que están por defecto.
- En la participación de más de 1 proyecto hay posibilidad de crear una base de datos diferente para cada proyecto o compartirla con un respectivo prefijo para las tablas de la nueva Wiki. Es recomendable usar una base de datos nueva para cada proyecto diferente, sin embargo para no duplicar el contenido existe una forma de hacer conexión entre las pagina de una Wiki diferente.

3.2.2 Estructuración del manual

Como segundo paso hay que crear una estructura del manual de usuario de la siguiente manera:

Una página principal con todos los proyectos o módulos a los que se les hará un manual de usuario, cada módulo debe ir referenciado a cada manual correspondiente, en este caso habrá un único proyecto.

Prefacio

Se puede hacer una nueva página “acerca de” sobre como navegar en el manual.

Ej - En este manual de usuario existe un vínculo al contenido en la página principal que pulsando sobre él, llevara a una estructura que explica todas las funcionalidades del software.

Índice

Un ejemplo seria:

- 1 Introducción
- 2 Cómo ingresar a moodle
- 3 Cómo salir de moodle
- 4 Composición de la plataforma
- 4.1 Usuario y menú principal
- 4.2 Calendario
- 4.3 Administración
- 4.4 Navegación
- 4.5 Panel central

Guía rápida sobre cómo usar las funciones principales del software

Sección para la resolución de problemas

FAQ

3.2.3 Gestión de roles

Una Wiki significa una ventaja para trabajar en grupo desarrollando alguna tarea específica que requiera la intervención de varias personas, cada una hace sus aportes y a su vez hay un control de los contenidos que se van generando. En la siguiente tabla se deja ver un ejemplo de la jerarquía de usuarios para la edición de contenidos.

Tabla 3 Asignación de roles

Usuarios \ Roles	Moderador	Editor
Freddysandoval	x	x
Usuario2		x
Usuario3		x

El usuario Freddysandoval tiene activado tanto el rol de Moderador como el de editor, esto quiere decir que puede crear y eliminar páginas de contenidos en la wiki así como aprobar los creados por usuarios con el Rol de editor [Usuario2] por ejemplo.

Para gestión apropiada de los Moderadores se sugiere instalar una extensión que protege los contenidos ya publicados y permite que el moderador apruebe cambios antes de ser publicados en la versión final que todos pueden ver (Edward Chernenko, n.d.).

3.2.4 Edición de contenidos

Una vez se ha instalado la herramienta, establecido la estructura que va a tener el manual como bien creado y repartido los roles para cada usuario, lo siguiente es empezar a crear el contenido que se quiere mostrar a los usuarios finales.

Se sugiere crear una página inicial con una lista de enlaces a todos los temas principales y para facilitar la navegación se puede agregar vínculos con el índice principal de contenidos en cada página de la documentación, esto se hace con el fin de que los usuarios y los editores puedan llegar al contenido que buscan o quieren editar rápidamente.

Para el caso de este ejercicio se establece que el usuario que se escogió como editor pueda crear páginas y nuevos contenidos solo cuando acceda desde su cuenta. Un ejemplo sería que haga cambios en la página principal.

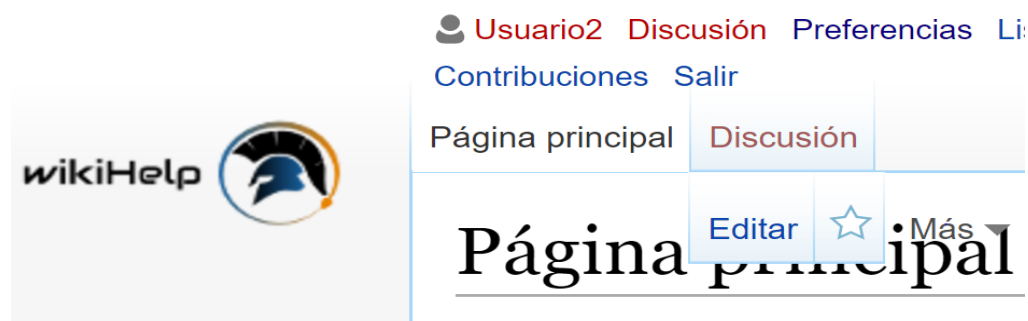


Imagen 12 Edición en página principal por Usuario2

El editor puede usar el lenguaje de marcas de la wiki por defecto en MediaWiki para crear los contenidos o Wikitext¹¹ correspondientes y dar formato a la página, también si tiene conocimientos de HTML también existe la manera de que los haga como considere más adecuado. Para terminar la edición solo tiene que guardar los cambios realizados, estos no se

¹¹ Wikitext - documento escrito en el lenguaje de marcado wiki (wiki markup language).

mostrarán inmediatamente a los usuarios finales sino que irán a un proceso de revisión de el que están encargados el o los moderadores.

Después de que el editor ha enviado su versión para revisar, el usuario con el rol de moderador tiene la tarea de revisar y aprobar o rechazar según sea necesario, para poder revisar las ediciones creadas por los editores tendrá que ir a la página (Especial:Moderation) en la wiki sería [<http://ingsistemas.unipamplona.edu.co/upwiki2/Especial:Moderation>] , esto abrirá la página que contiene la lista de todas las ediciones que esperan moderación (aprobar o rechazar).

Moderation

This special page lists all edits which await moderation (approval or rejection).

Pending moderation | [Rejected](#) | [Merged](#) | [Spam](#)

Abajo se muestran hasta **1** resultado entre el n.º **1** y el n.º **1**.

Ver (50 anteriores | siguientes 50) ([20](#) | [50](#) | [100](#) | [250](#) | [500](#)).

1. (diff) . . [Página principal](#) 10:33 . . (+28) . . [Usuario2](#) () [[Approve](#) [Approve all](#) . . [Reject](#) [Reject all](#)] . . [[Mark as spammer](#)]

Ver (50 anteriores | siguientes 50) ([20](#) | [50](#) | [100](#) | [250](#) | [500](#)).

Imagen 13 página de moderación de contenidos en la wiki

Si una edición es rechazada por defecto al cabo de dos semanas ya no podrá ser aprobada.

4 CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

El proceso de documentación para el usuario es un proceso importante en el desarrollo de software, por tanto son actividades que deberían trabajarse colaborativamente, como lo es el caso de la Universidad que lleva sus productos a muchas universidades y todos ellos requieren una descripción acertada para quienes usarán el software en sus distintas versiones.

En el desarrollo, la documentación de usuario es un proceso planificado, diseñado e implementado, ya que constituye un elemento de consulta para toda persona que va a usar el programa por primera vez o que trata de saber si el programa servirá a sus objetivos, igualmente para aquellos que quieren profundizar para un conocimiento avanzado, de esto depende gran parte de la usabilidad y el éxito futuro después de la instalación de una aplicación software. A pesar de esto los estándares no enseñan cómo hacer estas tareas, por lo que tener un procedimiento descrito como el propuesto es una gran ventaja a la hora de implementar este proceso al desarrollo de software.

Existen herramientas especializadas para la creación de manuales de usuario como es el caso de proyectos libres (Gnome, GNU) como DocBook y LinuxDoc que hacen el trabajo para documentar, en otros proyectos se hace necesario el uso de herramientas colaborativas de texto, editores de imagen y video, y por ultimo publicar los contenidos con algún servicio en internet.

La wiki como herramienta implementada es una opción escogida en este trabajo por su facilidad para realizar trabajo colaborativo, aunque no es la única sólo un buen análisis según

lo que se pretenda podrá ayudar a determinar la elección de la herramienta más adecuada para seguir la estrategia y crear un propio manual de usuario.

El mantenimiento de la documentación mediante el uso de herramientas colaborativas como las wikis, permite incluir a los propios usuarios del software como editores, que una vez encontrada desactualizada una documentación puede realizar aportes significativos en cuanto a nivel del lenguaje cotidiano del usuario como experto en el manejo de una funcionalidad particular del software.

Trabajo futuro

En este proyecto quedan pendientes aspectos como la integración de la herramienta usada para la creación de manuales de usuario en este caso una wiki con aplicaciones web, haciendo que la wiki sea la fuente de información actualizada de ayudas en línea de esas aplicaciones.

También se busca integración con otras herramientas de gestión de los procesos de software como es el control de versiones (Git por ejemplo), en el proceso de pruebas (Bugzilla) también la documentación en general no solo la parte de manuales de usuario.

5 REFERENCIAS

- Bachmann, F., Bass, L., Clements, P. C., Garlan, D., Ivers, J., Nord, R., ... Stafford, J. (2001). *Documenting Software Architectures: Organization of Documentation Package*. Carnegie Mellon University Technical Note.
- Bachmann, F., Merson, P., Architecture, S., Initiative, T., & Cmu, T. N. (2005). Experience Using the Web-Based Tool Wiki for Architecture Documentation. *Technology*, (September), Bachmann, F., Bass, L., Clements, P., Garlan, D.,.
- Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., ... Stafford, J. (2003). A Practical Method for Documenting Software Architectures. *In Proc. of the International Conference on Software Engineering (ICSE 2003)*.
- Edward Chernenko. (n.d.). Extension: Moderation. Retrieved from https://www.mediawiki.org/wiki/Extension: Moderation#Additional_anti-vandalism_tips
- MediaWiki. (2016). MediaWiki. Retrieved from <https://www.mediawiki.org/wiki/Download/es>
- skins. (2016). Retrieved from https://www.mediawiki.org/wiki/Category:All_skins
- Stallman, R. M. . (2002). *Free Software, Free Society: Selected Essays of Richard M . Stallman Introduction by Lawrence Lessig Edited by Joshua Gay*. *Computer Networks* (Vol. 25). <https://doi.org/10.1029/2004TC001693>
- Suárez, P., & Fontela, C. (2003). *Documentación y pruebas Antes del paradigma de objetos*.
- What Is Wiki. (2002). Retrieved from <http://www.wiki.org/wiki.cgi?WhatIsWiki>
- Wiki - EOI. (2012). Documentación en Software libre. Retrieved from http://www.eoi.es/wiki/index.php/Documentación_en_Software_libre#Docbook

6 ANEXOS

Anexo 1

Lo primero que hay que hacer es descargar e instalar de la página (MediaWiki, 2016) el archivo comprimido, en la página existe un manual de instalación dependiendo de la plataforma en la que se quiera instalar.

Se crea una carpeta nueva en la dirección `/var/www/html/` por ejemplo `[mywiki]`

En Linux `Mkdir /var/www/html/mywiki`

En la dirección `/var/www/html/mywiki` se descomprime el contenido descargado dentro de la carpeta, para descomprimir el archivo `.tar.gz` descargado se usa el siguiente comando.

```
tar xzvf archivo.ar.gz
```

Se configura la Wiki según en asistente de instalación indica el paso a paso.

Después de instalada la Wiki hay que descargar el archivo llamado `[LocalSettings.php]`. Es necesario colocarlo en el directorio donde se instaló la wiki, este archivo tiene todas las configuraciones de la Wiki así que siempre que se quiera colocar algo nuevo hay que editar este archivo.