PMWiki tutorial

En el MarkDown tutorial que hicimos explicamos de manera rápida la manera de escribir una página web con un formato desenfadado y sin tantas pretensiones. También publicamos acerca de **Marp**, un software libre capaz de realizar presentaciones y exportar a PDF a partir de un fichero hecho con MarkDown. Analizando el software de familias Wikis nos topamos con PMWiki que tiene una peculiar manera de escribir código "pseudo HTML" y por ello comenzaremos explicando las reglas de sintaxis y luego veremos como se instala y mantiene una página web con dicha solución en software libre, ¡adelante!

PMWiki text markup.

PMWiki utiliza *un lenguaje de marcado* a diferencia del paradigma actual del siglo XXI: «lo que usted ve es lo que obtiene» ("What You See Is What You Get" o por sus iniciales **WYSIWYG**). Esto último consiste en utilizar herramientas de modelado que presentan en tiempo real la apariencia del trabajo que hacemos sin importarnos cómo se logra ya que el "interprete" nos abstrae del código subyacente. El asunto es que nosotros nunca dejamos de estar conscientes que los ordenadores solo entienden de unos y ceros (y ni siquiera eso, ya que es una abstracción que nos hacemos: solo entiende con apagado o encendido, hay o no hay en el último de los casos).

Reglas básicas de edición en PMWiki.

Hay seis reglas básicas para editar las páginas web cuyo motor web es la PMWiki:

1. Para comenzar un párrafo solo basta con una simple línea en blanco.
2. Para comenzar una lista enumerada comenzamos cada línea con un caracter numeral"#"; una lista no enumerada cada línea utiliza un asterisco "*" *(numerada NO quiere decir ordenada, cuidado con los "falsos amigos" en las traducciones del inglés al castellano).*
3. Para un título principal comenzamos cada línea con signo de cierre de admiración "!" seguido de un espacio y luego el texto deseado. Cada subtítulo lleva dos o más signos de cierre de admiración "!!", "!!!", etc.
4. Para texto en cursivas lo encerramos con dos comillas simples ('') y para texto en negritas tres comillas simples (''').
5. Para realizar un enlace web en nuestra propia wiki lo encerramos con dobles corchetes rectos "[[*nombre*]]" donde *nombre* es el archivo que contiene nuestra página, *en realidad es un enlace relativo* (muy útil si luego queremos "mudar" nuestra wiki).
6. Si queremos enlaces web hacia otros sitios web también lo encerramos entre corchetes rectos doble y debemos escribir el enlace completo -vamos, dejad la pereza, escribid con todo y "**http://**". También sirve para otros tipos de enlaces como el usado para enviar correo electrónico "**mailto:**".

Párrafos y saltos de líneas.

Los párrafos se delimitan dejando una línea en blanco, por lo tanto todo lo que escribamos en varias líneas será presentado como un solo párrafo. Para los que amamos el utilizar la venta terminal para trabajar nos ofrece la ventaja de escribir con un máximo de 40, 80 o 90 caracteres de anchura lo cual nos permite una cómoda edición sin preocuparnos del ancho de pantalla que utilizarán nuestros usuarios.

Generalmente monitores con relación 16:9 que es apto para ver películas pero incómodo para ver párrafos muy anchos ya que dificulta seguir la línea que estamos leyendo -recordad el famoso papel pijama del siglo XX , *pero bueno, ese no es nuestro problema sino del usuario de la página, je, je, je ?*).

En la tabla anterior os mostramos de manera práctica los comandos y cómo se vería, *sin embargo es un artilugio de programación que le hicimos a Wordpress para ilustraros el temas, hay algunos detalles que no se ven igual en PMWiki* (lo mejor sería hacer un subdominio y correr una pequeña wiki allí para que veaís bien claro el resultado).

Listas.

Ya sabemos que para hacer una lista debemos comenzar con un asterisco "*" cada línea, y si acaso quiséramos asignarle una numeración automática debemos usar un numeral "#". Cualquier línea a continuación que NO comience con asterisco o numeral es implícito que finaliza la lista.

Acá ampliamos un poco con las sublistas: una línea que comienze con dos asteriscos (despúes de una línea que comienze con un asterisco) se visualiza como una sublista:

```
* Esta es una lista.  * Segundo elemento de lista.  ** Comienza sublista.
  ** Segunda línea de sublista.
```

Producirá el siguiente resultado:

- Esta es una lista.
- Segundo elemento de lista.
    - Comienza sublista.
    - Segunda línea de sublista.

Se pueden combinar:

```
* Lista  ## Sublista.  ## Dos.  ## Tres.
```

Salida:

- Lista.
    1. Sub lista uno
    2. Dos.
    3. Tres.

Algo que gusta mucho a los autores es la numeración romana o con letras del abecedario, ya sea en maypusuculas o minúsculas; esto se logra colocando las siguientes palabras claves luego de el o los asteriscos y/o numerales de cada línea (por supuesto, no se mostrará al usuario dichas palabras claves al momento de "servir" el artículo wiki):

- %roman%
- %ROMAN%
- %alpha%
- %ALPHA%

Ejemplos de códigos:

```
* %roman% Lista con números romanos.  * Línea dos en números romanos.  ##
 %ALPHA% Sublista con letras del abecedario en mayúsculas.  ## Segunda lí
nea de sublista.
```

Y arrojaría lo siguiente:

Estos "trucos" necesarios para mostrar en Wordpress listas y sublistas a "nuestra manera personalizada" tendrán su artículo aparte y será un complemento a nuestro tutorial HTML5 para no hacer ese tutorial más voluminoso de lo que ya es -y enseñar a usar Wordpress, que ni un temita le hemos dedicado a nuestro "motor" de blog.

Este concepto de palabras claves se repite si debemos forzar la numeración u orden: "**%item value= *número*%**".

También si queremos parar la numeración y reiniciarla solo debemos insertar una línea con la siguiente palabra clave: "**[==]**":

```
# %item value=7000% Línea siete mil.  # Siete mil uno.  [==]  # Reinicia numeración.
```

Salida:

7000. Línea siete mil.
7001. Siete mil uno.

1. Reinicia numeración.

Nota: el indentado que veís lo hace Wordpress para alinear las listas con diferentes numeraciones, se agradece para nosotros los maniáticos del orden.

Listas con párrafos explicativos.

Para cada elemento de una lista podemos anexar un párrafo a continuación para ampliar o definir un concepto o descripción. El estilo indica que ha de ser lo más breve y conciso posible para mantener el panorama de la lista en sí misma. Dijimos que cualquier línea a continucación que NO comience con asterisco o numeral significa el fin de la lista pero he aquí que si una línea comienza con un espacio *se considera esa línea como perteneciente al elemento de la lista que lo precede* ***pero no se enumera -no es un elemento de la lista en sí-.*** Para lograr el indentado se debe finalizar la línea elemento de lista con "\\" (que aunque no es esctrictamente necesario, mejora el aspecto).

Un ejemplo:

```
'''Grandes venezolanos'''  * Simón Bolívar, Caracas 1783, Santa Marta 1830.\\   ''«Llamarse jefe para no serlo es el colmo de la miseria»''.  * Francisco de Miranda, Caracas 1750, Cádiz 1816.   ''«¡Le Venezuela est Blessée au coeur!»''.  * José Antonio Páez.
```

Produce lo siguiente:

Grandes venezolanos:

- Simón Bolívar, Caracas 1783, Santa Marta 1830.

«Llamarse jefe para no serlo es el colmo de la miseria».

- Francisco de Miranda, Caracas 1750, Cádiz 1816.

«¡Le Venezuela est Blessée au coeur!».

- José Antonio Páez.

Línea horizontal.

Si escribimos el comando "**----**" (cuatro o más guiones seguidos) en una sola línea de texto nos produce una línea horizontal gráfica que sirve para separar párrafos o denotar importancia como una advertencia -o el uso que queráis-.

Texto sin formato, tal cual.

Supongamos que en nuestra PMWiki queremos hacer un tutorial en castellano sobre todo esto que estamos explicando acá vía Wordpress: para indicarle *que no proceso nuestros código ejemplos y los muestre tal cual, cómo código que son* debemos "escapar" el texto de código con "[@ texto código @]". Puede ser utilizado para un párrafo completo o para una o más palabras en un párrafo.

Para un "escapado" estricto utilizamos "[= texto código =]", la diferencia esta que con arroba se respetarán los saltos de líneas y espacios pero con el signo de igualdad se escribirá todo junto como un párrafo .

Texto con fuente monoespaciada.

Si además necesitamos una fuente monoespaciada, es decir, un tipo de fuente cuyas letras todas y cada una tienen el mismo ancho simplemente debemos comenzar la línea con un espacio (que no esté después de una lista, ojo con eso). Imaginad (o recordad) los famosos cuadernos cuadriculados para matemáticas, una letra por retícula.

Tablas.

Para las tablas emplearemos el caracter doble "tubería" o "||":

- La primera línea con doble tubería seguido del tipo de borde: "|| border=1".
- La segunda línea con la primera fila de la tabla y cada elemento de columna separado con doble tubería.
- Si es un encabezado de tabla, luego de doble tubería un signo de admiración de cierre.

Una tabla sin borde se logra con "border=0", de uno en adelante produce un borde doble con un sombreado que aumenta según aumenta el valor entero que le demos.

Existe una sintaxis alterna para tablas más avanzadas **pero en honor a la verdad dicha sintaxis es igual de compleja que escribir en código HTML. Pensamos** que solamente vale la pena para un cuadro de navegación como el siguiente:

```
(:table border=1 width=30% align=right bgcolor=#cccc99 cellspacing=0 : )
 (:cellnr: )  '''Páginas de buscadores'''  (:cellnr: )  *[[https://duckdu
ckgo.com/ | DuckDuckGo]]  *[[https://fr.wikipedia.org/ | Wikipédia L'ency
clopédie libre]]  (:tableend:)
```

En esta imagen podemos ver la *tabla avanzada* mezclada con los ejemplos últimos, debajo de la imagen coloamos TODO el código completo que produce ese resultado en PMWiki:

```
(:table border=1 width=30% align=right bgcolor=#cccc99 cellspacing=0 :)
(:cellnr:)  '''Páginas de buscadores'''  (:cellnr:)  *[[https://duckduckg
o.com/ | DuckDuckGo]]  *[[https://fr.wikipedia.org/ | Wikipédia L'encyclo
pédie libre]]  (:tableend:)  '''Grandes venezolanos'''  * Simón Bolívar,
Caracas 1783, Santa Marta 1830.\\   ''«Llamarse jefe para no serlo es el
colmo de la miseria»''.  * Francisco de Miranda, Caracas 1750, Cádiz 1816
.   ''«¡Le Venezuela est Blessée au coeur!»''.  * José Antonio Páez.  ||
border=0  ||! head 1 ||! head 2 ||! head 3 ||  || cell 1 || cell 2 || cel
l 3 ||  || border=1  ||! head 1 ||! head 2 ||! head 3 ||  || cell 1 || ce
ll 2 || cell 3 ||  || border=2  ||! head 1 ||! head 2 ||! head 3 ||  || c
ell 1 || cell 2 || cell 3 ||  || border=3  ||! head 1 ||! head 2 ||! head
 3 ||  || cell 1 || cell 2 || cell 3 ||
```

Nota: la tabla avanzada se colocará siempre a la derecha, tal como instruimos en código pero al mismo nivel de los elementos que tenga a continuación -uno al lado del otro-

Formato de caracteres.

Otros comandos nos permiten dar formato al texto para representar fórmulas, citas, etc

```
* Texto con '^superíndices^'.  * Texto with '_subíndices_'.  * Texto {-ta
chado-}.  * Texto {+subrayado+} (poco utilizado, se confunde con hipervín
```

```
culo).  * Texto [+grande+], [++más grande++].  * Texto  [-pequeño-], [--m
ás pequeño--].
```

En la siguiente sección colocamos una imagen tomada del **WikiSandBox**: es un área donde podremos libremente practicar nuestros códigos, podremos hacer y deshacer sin problema alguno, programamos y visualizamos resultado *mientras agarramos práctica, la idea es luego escribir rápidamente artículos completos y publicar de una vez sin mayor dilación.*

"**Sandbox**" significa "cajón con arena" que nunca falta en los parques de los Estados Unidos y donde colocan a los pequeños y pequeñas para que desarrollen sus habilidades motoras sin peligro alguno; acá en Venezuela hacemos algo mucho mejor: los llevamos a las playas maravillosas de nuestro país.

Alineación de texto, color de fuente.

Por defecto el texto está alineado a la izquierda, pero podemos centrarlo comenzamos la línea con "**%center%**" y "**%right%**" para alinearlo a la derecha.

Para que el texto "flote" a la derecha sobre el siguiente párrafo que haya a continuación iniciamos con "**%rfloat%**" y si lo queremos en un marco "**%rframe%**".

Para aplicar color simplemente escribimos el color, en inglés, encerrado entre signos de porcentaje. Esto se aplicará por defecto al resto del párrafo a partir del código, si necesitamos solamente colorear una palabra escribiremos "**%blue% palabras a colorear %black%**". *Por defecto el color de fuente es negro, por eso escribimos %black% pero eso no es necesariamente cierto, todo depende del estilo general de la PmWiki (tema avanzado).* **Lo correcto para restablecer el color es cerrar con doble signo de porcentaje "%%".**

También se pueden combinar los comandos, por ejemplo centrado y en color rojo: "**%center red%".**

Alcance de los estilos.

No necesariamente se deben colocar los códigos al inicio de un párrafo, por ejemplo, a mitad de un párrafo podremos escribir "**%p red%** indicando que el párrafo, representado por la letra "p" sea rojo "red" sin importar dónde lo ubiquemos: al principio, en el medio o al final. Similar comportamiento aplica para otros elementos:

- Imágenes: %img ... (otros comandos)%

- Texto de código: %pre (otros comandos)%
- Lista: %list (otros comandos)%
- Elemento de lista: %item (otros comandos)%

Estilos con nombres en castellano.

Podremos definir clases de estilos, definidos por nosotros mismos. En mejor uso es poder escribir en castellano, especialmente para aquellos u aquellas que no dominan ese idioma. Para ello al comienzo de un artículo podríamos escribir lo siguiente:

```
%define=amarillo color=yellow% %define=fondo_amarillo bgcolor=yellow% %define=azul color=blue% %define=fondo_azul bgcolor=blue% %define=rojo color=red% %define=fondo_rojo bgcolor=red% %amarillo%Conocerás los principios básicos%% de la programación en PHP como su sintaxis, estructuras de control, etc. %azul%Además explorarás un nutrido conjunto de características%% y funcionalidades, las necesarias para desarrollar la mayoría de las aplicaciones, %fondo_rojo%como el acceso a la base de datos%%, el sistema de archivos, etc.
```

Así entonces será más fácil recordar los comandos, pero supone escribir guiones al principio de cada artículo. **Es mejor definir estilos globales, que funcionen en toda la página PmWiki alojada, eso es avanzado y más adelante lo estableceremos en la configuración de instalación.**

Caracteres especiales.

Pues con nuestros teclados o copiando y pegando podemos escribir los caracteres "especiales": las vocales acentuadas, el símbolo de la moneda euro €, etcétera y para ello rigen las mismas normas del HTML: la "á" se escribiría "**á**" el símbolo del €: **€**.

Imágenes.

Por increíble que parezca, para mostrar una imagen en nuestra PmWiki simplemente colocamos el enlace hacia la imagen web... ¡y listo! (en Wordpress es igual, aunque no lo creáis, pero este tutorial es de PmWiki y no otra cosa, disculpad).

El meollo del asunto es que normalmente gustamos de darle formato a las imágenes, texto alterno si el navegador no puede mostrar imágenes -o las tienen bloqueadas de alguna manera-, títulos, tamaños, bordes, etc. Comencemos a realizar una lista con las principales características:

- Comenzemos por establecer que una imagen ocupa una línea completa, *imaginad que estamos escribiendo un párrafo.* Luego cuando cojamos práctica podremos mezclar imagen entre párrafos para lograr bonitos acabados, verbigracia, como una enciclopedia merece; *pero vayamos aprendiendo poco a poco.*
- Simplemente una imagen, sin más, colocamos una URL completa con alguno de los siguientes formatos de imágenes: gif, jpg, jpeg, png, svg, svgz. Dicha imagen *debería residir en nuestro servidor por principios de ética de no "robar" el ancho de banda de los demás sitios web.* **Aunque PmWiki permite subir imágenes, dicha función, por seguridad, viene desactivada por defecto. En la sección de instalación y configuración tomaremos de nuevo el tema.**
    - Ejemplo de imagen: http://pmichaud.com/img/misc/pc.jpg
    - La anterior imagen está alojada en el servidor web del autor de PmWiki, el Doctor Patrick Michaud, la imagen fue tomada de Flickr y tiene licencia "Creative Commons".
- Cuando uno ubica, sin hacer click, el puntero del ratón (y nos disculpáis la redundancia) sobre una imagen web por más de 500 milisegundos (generalmente) nos aparece una ventanita con un texto que describe la imagen: para lograrlo en PmWiki debemos escribir *a continuación y sin dejar espacio alguno,* encerrado entre comillas dobles el texto deseado. **También este texto es mostrado si por razón alguna el navegador NO puede mostrar la imagen,** y los navegadores web modernos que narran las páginas web también toman este texto para los discapacitados visuales.
    - Ejemplo de "tooltip": http://pmichaud.com/img/misc/pc.jpg"CLIPS"
- Si queremos establecer un título a la imagen, después de las comillas dobles y sin espacio alguno insertamos un caracter "tubería" "|" e inmediatamente el texto deseado. *En honor a la verdad si acepta espacios entre el caracter tubería pero recomendamos escribir todo junto para ayudarnos a recordar el proceso (y pulsar **menos** teclas para escribir más rápido).*
    - Ejemplo de título a imagen: http://pmichaud.com/img/misc/pc.jpg"CLIPS"|"Clips coloridos"
- Si queremos que la imagen en sí sea un enlace web debemos prestar mucha atención: debemos encerrarla entre corchetes rectos dobles *pero dejando fuera el título de imagen, si es que la hubiere.*
    - Seguimos con el mismo ejemplo, ahora más larga la línea:
    - [[http://pmwiki.org/|http://pmichaud.com/img/misc/pc.jpg"CLIPS"]]|"Clips coloridos"
- Al igual que un texto, podemos alinear la imagen al centro o derecha colocándole de principio de línea los comandos **%center%** o **%right%** (por defecto se alinea a la izquierda):
    - %center%[[http://pmwiki.org/|http://pmichaud.com/img/misc/pc.jpg"CLIPS"]]|"Clips coloridos"
    - Ahora vamos viendo el porqué recomendamos eliminar los espacios, va larga la cosa. :-)
- Hasta ahora la imagen que queremos mostrar es tal cual, **sin ninguna modificación al**

tamaño. *Rogamos encarecidamente que subáis imágenes ligeras, en formato jpg preferiblemente y que midan menos de 50 mil bytes.* Para redimensionar debemos usar:

- A un ancho específico: **%width=150px%** (a 150 píxeles de ancho).
- A un alto específico: **%heigth%=100px%** (a 100 píxeles de alto).
- A una miniatura (66 x 32 píxeles, formato gif): **%thumb%**.



- Intentamos seguir con el ejemplo planeado pero las múltiples combinaciones impiden que PmWiki redimensione la imagen, bien puede ser un excepción de PmWiki o nosotros desconocemos la sintaxis correcta *pero nos decantamos -humildemente- por el error "bug" de PmWiki.*

- Para que al hacer click en la imagen se abra en una pestaña o ventana nueva debemos encerrar todo el enlace y sus opciones adicionales entre **"%newwin% *enlace_web* %%"** *teniendo la precaución de colocarlo todo junto y justo antes de los corchetes dobles.* Si quisieramos centrar la imagen primero debemos escribir %center% y luego el resto del comando *y no al revés porque no lo tomara de forma debida y esperada.*
- Un interesante efecto para un párrafo es insertar la imagen a la izquierda o a la derecha de un párrafo que la describa o hable de ella, para lograr esto simplemente colocamos **%rframe%** o **%lframe%** ***antes del código de la imagen y luego al finalizar el código en otra línea el párrafo, sin dejar líneas vacías de por medio.***
  - %rframe%https://www.ks7000.net.ve/wp-content/uploads/2017/05/PmWiki-logo-1.jpg ***párrafo***



  - De nuevo, las opciones de redimensionamiento tampoco logramos hacerlas funcionar con **%rframe%.**
  -

- Existen muchísimas más opciones para elt ratamiento de imágenes *pero escapan a la simplicidad que queremos lograr y se aproximan más a la sintaxis de CSS y HTML, y es* ***justo lo que queremos evitar, escribir en esos dos lenguajes de marcado.***
- Las imágenes se pueden combinar con tablas pero de nuevo, no son nada sencillo y queremos evitar su uso.
- ***PmWiki es tan poderoso como queráis, no lo subestiméis, al modificar "config.php" tendremos cientos de personalizaciones, pero mantengamos la simplicidad, por favor -no todos nuestros usuarios serán tan avanzados en programación, con que aprendan unos cuantos comandos bastará para que escriban en nuestra Wiki.***

Historia de PmWiki.

La palabra **wiki** proviene del idioma hablado en la isla de Hawaii, quincuagésimo estado federal de los Estados Unidos de América. Significa rápido y **wiki wiki** súper rápido (ya en los años 80 acá en Venezuela vendíamos un tinte para ropa llamado "wiki-wiki": no teníamos dinero para comprar pantalones nuevos y los pintábamos de negro y listo, "pantalones nuevos" rápidamente).

Las páginas web que tienen un motor **wiki** son como cualquier otra página web pero tienen dos características únicas:

- Tienen un botón con la palabra "editar" para que cualquiera colabore en ellas (y podemos subir o bajar los requerimientos de seguridad en la medida que nos sintamos cómodos).
- No se necesita de conocimientos profundos sobre desarrollo web, con unos cuantos comandos como los que vimos arriba en el resumen de 6 pasos bastan para que se comienze a plasmar el aporte de cada quien.

Autor de PmWiki.

El caso concreto de PmWiki el Doctor Patrick Michaud (la letra "d" NO se pronuncia) es el autor con más de 40 años de experiencia en computación (graduado con "Post High Degree" en la University of Southwestern Louisiana -aquí le decimos Doctorado-) y podéis acceder a su página web http://www.pmichaud.com/ donde, faltaría más, faltaría menos con el motor PmWiki donde mantiene un resumen de su vida profesional y formas de contactarlo ya sea por correo eletrónico o redes sociales.

Filosofía de trabajo de PmWiki.

Fueron establecidas durante el desarrollo de PmWiki y aunque el Dr. Michaud comparte muchas de ellas no son exclusivas de su persona, *de hecho mantienen un foro abierto a* nuevas ideas y sugerencias *para mantener activo este software a los requerimientos modernos de la humanidad.*

Los pilares son:

1. Preocupados sobre los autores más que por los lectores: buenos autores son difíciles de conseguir y "mantenerlos" en el proyecto priva sobre cualquier lector de la enciclopedia (crudo pero es la realidad).
2. PmWiki *no trata de reemplazar al HTML*: ya hemos hablado sobre esto, se enfoca en ser algo simple y sencillo para los autores, sin embargo tiene poderosos sustitutos que pueden llegar a realizar casi lo mismo que el HTML *pero no es nunca ese su objetivo*.
3. Pragmatismo sobre todo: se busca la utilidad por encima de la moda, algo bien

estructurado en vez de ir a lanzarse a la primera por cualquier tendencia.
4. Soporte colaborativo para el mantenimiento público de páginas web: autores y grupos de autores pueden ir construyendo un sitio para luego llegar a un punto de madurez y empezar a proteger con contraseñas ya sean páginas individuales o secciones o grandes secciones que busquen preservar los títulos, subtítulos e incluso párrafos e imágenes mientra otras áreas están libres y abiertas para captar nuevas ideas o registrar el simple pasar del tiempo. Precisamente éste es un ejemplo: un registro de cambios de un artículo (NO el registro de cambios de la página en sí) debería estar protegido al cabo de cierto tiempo, días o semanas, proque eso es historia y ya no cambiará (incluso se pueden dejar las referencias abiertas en pie de página si alguien quiere agregar algo al tema).
5. **Fácil de instalar, configurar y mantener:** *del dicho al hecho, las siguientes secciones hablan sobre ello.*

Instalación de PmWiki.

Primero debemos descargarlo de forma comprimida desde PmWiki, solo pesa menos de medio megabyte, la versión estable 2.2.98 -ya os dijimos: PmWiki es sencillo y simple pero muy útil-.

Como un servicio a la comunidad por acá también la ponemos a la orden, si la queréis descargar desde acá.

Se ofrece paquetes adicionales para diferentes idiomas en esta dirección de la misma PmWiki.org pero deja mucho que desear *por ahora para nuestro trabajo la vamos a dejar así, en purito inglés pero no creáis que nos hemos rendido con el castellano, de hecho tenemos en mente unas ideas laboriosas por realizar y una vez terminemos la publicaremos por acá.*

Una vez descargado lo descomprimimos en nuestra carpeta **/var/www/html** -en nuestro caso usamos un servidor web Apache2 con PHP5 en nuestro ordenador en VirtualBox- y debemos darle los derechos adecuados de escritura para que se puedan grabar nuestras preferencias.



Se recomienda crear una carpeta llamada **pmwiki** en la carpeta pública de nuestro servidor web y redirigir debidamente modificando el .htaccess (o por otros métodos). No es la tarea de esta entrada y tomando en cuenta que tenemos un servidor de pruebas y por propósitos didácticos NO LO HAREMOS ASÍ sino que simplemente colocaremos los ficheros y directorios como si el servidor fuera dedicado a retribuir una sola página web.

Ficheros y carpetas de PmWiki.

Una vez hayamos copiado los archivos tendremos la siguiente estructura que pasamos a traducir directo del propio tutorial en inglés:

- README.txt: un documento introductorio *y contiene la licencia de uso GNU:*
  - https://www.gnu.org/licenses/gpl.txt
- pmwiki.php: el archivo principal, el hace las veces del "index.php" en nuestro servidor Apache2.
- local/: Carpeta que contiene los guiones de configuración local.
- cookbook/ : Carpeta con"recetario" con programas agregados que enriquecen a PmWiki (suplementos).
- docs/ : Documentación y guiones de ejemplo.
- pub/ : archivos de acceso público.
  - pub/css/ : Archivos con hojas de estilo en cascada de los temas (suplementos).
  - pub/guiedit/ : (¡aún no hemos estudiado este directorio!).
  - pub/skins/ : "temas" para cambiar la apariencia y comportamiento de PmWiki.
- scripts/ : Guiones que son partes de PmWiki.
- wikilib.d/: Paquete de páginas predeterminadas de PmWiki.

Por ahora esos son los ficheros y directorios iniciales, los cuales, por seguridad, solo necesitamos que tengan derecho de lectura más no de escritura. **Para poder guardar nuestras configuraciones y futuros artículos debemos tener derechos de escritura en una carpeta especial llamada "wiki.d/".**



¡Atención! no confundir con la carpeta wikilib.d

- Dentro de esta carpeta **wiki.d** se guardarán nuestros artículos con el prefijo **"PmWiki."** más el nombre del título que le pongamos al crearla.
- Junto a estos archivos automáticamente PmWiki llevará de manera sincronizada otros ficheros donde se registran los cambios hechos, la configuración de la barra lateral izquierda e incluso un índice de páginas. *No debemos preocuparnos por estos archivos, excepto respaldarlos regularmente fuera de nuestro sitio web y/o servidor de producción* (si respaldamos en nuestro ordenador podremos llevar un respaldo adicional *junto con un control de cambios* al utilizar GitHub el cual es sin costo para proyecto públicos -proyectos privados se paga mensual en US$-).
- El otro directorio que debe tener derechos de escritura *si es que llegamos a habilitarlo* es la

carpeta "**uploads**" que viene a albergar los ficheros que subamos a nuestro servidor. Se le harán las mismas consideraciones del punto anterior.

Si nuestro servidor permite acceso vía SSH por ventana terminal usaremos el comando **"chmod"** para otorgar permisos de escritura, si es un servidor web compartido *generalmente tenemos acceso vía softwfare CPANEL que permite lanzar un navegador de archivos y otorgar derechos de escritura a esas dos carpetas.* También recordemos que el comando **"chmod"** funciona en muchos programas clientes FTP, así que cuando "subamos" los archivos de PmWiki podríamos cambiar de una vez los derechos de escritura.

Dado el caso que NO hayamos configurado lo anterior y nos decidimos por ejecutar directamente de una el fichero principal **pmwiki.php** éste nos lo recordará amablemente con un mensaje de que no podemos seguir adelante.

Ejecutando por primera vez pmwiki.php

Pues bien, acá se mostrará la página de bienvenida con tutoriales e indicaciones -en inglés- de todo lo que explicamos acá en idioma castellano.

Estableciendo "index.php" o "index.html"

Dependiendeo de como tengáis configurado vuestro servidor, necesitaréis un archivo llamado "index.php" o si tenéis configurado tratar los archivos con extensión .html como .php debeis escribir en él lo siguiente:

[cc lang="php" tab_size="2" lines="60"][/cc]

Modificando "config.php".

Lo que nos toca por hacer es configurar el archivo **config.php** en la carpeta "**local/**". Dicho archivo es de solo lectura *pero debemos tener permisos de escritura para copiarlo desde la plantilla llamada* **"sample-config.php"** (carpeta **"docs/"**) la cual modificaremos a nuestro gusto y conveniencia: en nuestro ordenador donde hayamos descargado y descomprimido **PmWiki** copiamos el archivo con el nombre **"config.php**" *lo modificamos* para luego subirlo (vía **https** o **sftp**) a nuestra carpeta **"local/"**.

Este fichero contiene las configuraciones comunes *pero vienen inactivas gracias a un caracter numeral al principio de cada línea de comando, eliminando dicho caracter* ***las "descomentamos" y ponemos a nuestro servicio.***

- La primera línea que modificaremos es el título de nuestra página, por defecto reza "**$WikiTitle = 'PmWiki';**" cambiando solamente lo que está entre comilla simples.
- La segunda línea a cambiar permite cambiar nuestro logotipo a una imagen tipo gif de 32 píxeles que diseñemos y con el nombre que querramos, cambiamos la siguiente línea QUE DEBEMOS DESMARCAR como comentario (fijaos en la ruta donde vamos a subir dicho archivo):
    - $PageLogoUrl = "$PubDirUrl/skins/pmwiki/pmwiki-32.gif";
- ¡Muy importante! la contraseña administrativa en la línea siguiente la cambiaremos a nuestro criterio:
    - $DefaultPasswords['admin'] = pmcrypt('secret');
- Si quisieramos activar la subida de archivos, modificar la línea que activa (descomentarla) y fijar una nueva contraseña a nuestra conveniencia (y descomentarla también):
    - $EnableUpload = 1;
    - $DefaultPasswords['upload'] = pmcrypt('secret');
- PmWiki permite unos botones gráficos para darle un pequeño toque "colorido" a nuestra página web, si deseais lo activaís descomentando la siguiente línea:
    - $EnableGUIButtons = 1;
    -

¡LISTO! Guardamos y copiamos a su ubicación final y vemos los cambios recargando la página.

Seguridad en PmWiki.

Al modificar **config.php** establecimos nuestra contraseña como administrador, nuestra llave para crear, modificar o eliminar contraseñas, es el más alto nivel efectivo en nuestro sitio. **La reservamos para nosotros, los programadores y encargados del mantenimiento y funcionamiento de la página web en sí** ya que la mayor del tiempo serán los autores los que estarán a cargo de proteger sus propios trabajos.

El enfoque que utilizaremos, siempre pragmáticos, es de ir de menos a más, según nuestras necesidades, analizemos:

Grupo de 1 a 3 autores.

Siendo un grupo pequeño solo se necesita que a *nivel global* todos los botones de edición de página tengan una simple contraseña. Para lograr esto debemos *descomentar* y modificar ***(o agregar)*** la línea siguiente en **/local/config.php**

```
$DefaultPasswords['edit'] = pmcrypt('nuestra_contraseña_de_edicion');
```

Acá no hay más complicaciones, solo es cuestion de comunicarse entre ellos y verbalmente dirimir sus diferencias, en teoría no debería haber mayor problema *y si lo hubiera nosotros como administradores deberemos tener la última palabra y "reiniciar" la contraseña a nivel global para excluir a algún miembro de la edición de artículos -caso extremo-.* Luego veremos que tal vez tengamos unas tareas adicionales que realizar, tened paciencia y os explicamos, vamos a continuar con la idea de grupos.

Grupo de 3 a 12 autores.

Manejar a esta cantidad de personas la cosa se complica un poco. Tratar de reunirlos todos a la vez va a ser difícil y si se lograra se nos va a ir el tiempo en reuniones grupales, lo cual es mucho trabajo para el poco dinero que nos pagan por la tarea. ?

El mejor enfoque en este caso es establecer nosotros mismos **normas de publicación** (grupo de páginas) de caracter privado por lo que tendremos que establecer contraseñas para poder leer este particular grupo de páginas. Establecidas las normas, ya saben a qué atenerse. Una vez "tamizados" los autores podremos elegir a dos sub administradores que nos ayuden en nuestro trabajo, pero *¿cómo lograr esto sin dar nuestra contraseña de edición a los subadministradores?* A nivel global se pueden establecer matrices de contraseñas **(esto ya es lenguaje PHP en sí)** de nuevo en el ya famoso archivo **config.php**:

```
$DefaultPasswords['read']    =    array(pmcrypt('alfa'),    pmcrypt('beta
'));  $DefaultPasswords['edit'] = pmcrypt('beta');
```

"alfa" y "beta" serán dos contraseñas distintas, para elevar un autor a subadministrador *(ojo, esta figura no está canonizada en el sistema PmWiki, la estamos "inventando" en este artículo práctico)* pues le daremos una nueva contraseña que será "beta". **Es más, si son dos subadministradores, como proponemos ,nada nos impide generar una contraseña "gamma":**

```
$DefaultPasswords['read']    =    array(pmcrypt('alfa'),    pmcrypt('beta
'),          pmcrypt('gamma'));                    $DefaultPasswords['edit
'] = array(pmcrypt('beta'), pmcrypt('gamma'));
```

Luego veremos que tal vez tengamos unas tareas adicionales que realizar, tened paciencia y os explicamos, vamos a continuar con la idea de grupos.

Grupo de autores de hasta 100 autores.

Acá la cosa se pone "color de hormiga": ya en este punto debemos habilitar *usuario y contraseña* (léase "login" y "password") para todos y cada uno de los usuarios a fin de determinar quién hizo qué y tomar las medidas correspondientes. ¿Recordáis la figura de subadministradores? **Pues bien, ¡vamos a necesitar entre 3 y 6!** Esto es así porque el mantener cuentas y contraseñas de usuario se va a llevar bastante trabajo y necesitamos

El primer paso para habilitar esta característica es agregar (o descomentar) la siguiente línea en **/local/config.php:**

```
include_once("$FarmD/scripts/authuser.php");
```

Demás está decir que ya habremos establecido nuestra contraseña de "admin", lógicamente, en vez de la que trae por defecto la instalación de PmWiki (un *hacker* -buenas intenciones- o un *cracker* -malas intenciones- lo primero que haría sería intentar entrar con esa contraseña por defecto).

La manera de registrar a los usuarios es editar una página llamada **pmwiki.php?n=SiteAdmin.AuthUser**, es decir, al guión principal llamado **pmwiki.php** le "pasamos" en la barra de direcciones un signo de cierre de interrogación y solicitará nombre de usuario y contraseña al editar (usaremos la superclave de **admin**) y agregaremos cada cuenta con la siguiente sintaxis:

nombre_de_usuario: (:encrypt *contraseña*:)

Guardamos la página y la contraseña se mostrará encriptada. Dichas cuentas, por manía de orden, la colocamos justo después del subtítulo "**Login accounts**" y más abajo veremos el subtítulo "**Authorization groups**" donde podramos definir los grupos de usuarios que inventamos, a saber: autores y subadministradores.

```
@autores: Kevin, Jimmy  @subadministradores: Kevin
```

Al crear este tipo de grupos de usuarios se nos facilita enormente la tarea de administración si invertimos el enfoque: nombramos a cada usuario a los grupos que puede pertenecer, el ejemplo anterior *pero por usuarios:*

```
Kevin: @autores, @subadministradores  Jimmy: @autores
```

¿En qué nos beneficia todo esto? En que en cada página podemos definir cuales grupos tienen acceso *e incluso podemos delegar -con sumo cuidado- de poder editar la mismísa página* **SiteAdmin.AuthUser**. A primer vista esto parecería una locura pero es que para definir cuentas de usuarios y grupos de usuarios la otra manera de hacerlo permitir editar el fichero **config.php** y esto no es para que lo haga todo el mundo, cualquier cosa podría salir mal y tendríamos nosotros mismo que enmendar el capote.

Luego veremos que tal vez tengamos unas tareas adicionales que realizar, tened paciencia y os explicamos, vamos a continuar con la idea de grupos.

Grupos de 100 o más autores.

Manejar tal cifra de personas ya no es posible hacerlo así, "de manera manual", tendremos que recurrir al uso de protocolos y normas adecuados y que manejen bases de datos para llevar cuenta de todo. **PmWiki** ofrece acceso mediante el protocolo LDAP *pero escapa al alcance de este tutorial.* También existe apoyo para **htpasswd:** el cual utiliza el servidor **Apache2** pero no supone ventaja alguna porque al igual que **PmWiki** se basa en texto plano -incluso encriptado- y no utiliza base de datos *además* deja por fuera a **Ngix** que es otro poderoso servidor web, así que no le vemos futuro a esa alternativa.

Además, para manejar tal cantidad de usuarios, se espera una gran cantidad de artículos y por ello debemos hacer uso de un manejador de base de datos que no exija demasiado recursos de cómputo: **SQLite** es el elegido para ello. Para habilitar su uso debemos descargar y ejecutar este guion creado por Petko Yotov con el cual creamos la base de datos, tablas, índices y relaciones y una vez hecho esto si que debemos modificar de nuevo el consabido **config.php** en alguna de sus primeras líneas lo siguiente:

[cc lang="php" tab_size="2" lines="60"]include_once("$FarmD/cookbook/sqlite.php");
$WikiDir = new PageStoreSQLite($WorkDir.'/pmwiki.sqlite.db', 1);
$WikiLibDirs = array(
&$WikiDir,
new PageStore('wiki.d/{$FullName}'),
new PageStore('$FarmD/wikilib.d/{$FullName}')
);[/cc]

El fichero **pmwiki.sqlite.db** será entonces el único que conservará las página que creemos y será el que dabamos respaldar llegado el momento necesario. *Esto apenas es un resumen ya que todo programador sabe muy bien que no hay base de datos "rebeldes" sino mal administradas. Por ejemplo, de tanto en tanto deberemos ejecutar la orden* ***pmwiki.php?action=vacuum****; también podremos establecer un valor automático para que ejecuta esta acción por nosotros si lo establecemos en* ***config.php:***

```
$SQLiteAutoVacuum = 300;
```

Incluso si le colocamos el valor de cero (no ejecutar), siempre podremos hacerlo cuando necesitemos con **?action=vacuum.**

Tareas adicionales.

Pues que varias veces nombramos unas dichosas "tareas adicionales" que debíamos realizar para "cerrar el círculo de seguridad" para ello los siguientes datos:

- Las páginas y grupos de páginas les pueden ser establecidas, individualmente,
    - Contraseña de lectura *read.*
    - Contraseña de escritura *write.*
    - Contraseña de acceso a configurar las dos anteriores mediante *attr.*
- Para modificar las contraseñas de lectura y escritura a una página debemos adicionar "?action=attr" al final en la barra de direcciones y así introducir nuestra contraseña de administrador (admin).
- Para modificar las contraseña de lectura y escritura a un **grupo de páginas** debemos adicionar al final en la barra de direcciones lo siguiente (susitutid "GroupName" por el nombre del grupo de páginas:)
    - "?n=GroupName.GroupAttributes?action=attr"
- Las contraseña son visibles para no pedir confirmación pero al ser guardadas serán debidamente encriptadas (cuidado con las miradas encima por encima de tus hombros).
- En cada campo -lectura y escritura- **de una página *o grupo de páginas*** podremos:
    - Agregar una o más contraseñas simplemente separándolas por un espacio.
    - Agregar uno o más grupos nombrandolos precedidos de una arroba -ejemplo *@subadministradores*- y separándolos por un espacio entre ellos.
    - Si queremos que todos los miembros de un grupo tengan acceso usaremos "**@nombre_grupo: ***".
    - Si de ese grupo necesitamos exceptuar un usuario -que pertenece a ese grupo-:
        - "**@nombre_grupo: *, -nombre_usuario**".
    - Establecer uno o más usuarios con la palabra clave **"id:"** y a continuación los nombres de los usuarios deseados separados por comas.

- Para incluir todos los usuarios: **"id:*"**
- Dejar la contraseña en blanco *no guarda los cambios para ese campo.*
- Si queremos quitar la(s) contraseñas debemos usar la palabra clave "**clear**" solamente.
- *Evidentemente ningún usuario podrá llamarse "clear" (no hemos probado si admite tal usuario)-*
- Si queremos dejar sin contraseña, así esté establecida a nivel global usaremos "@**nopass**" solamente.
- Si al contrario, solo queremos que nosotros como administrador seamos los únicos con derechos sobre una página o grupo de páginas usaremos la palabra clave "@**lock**" solamente.
- Si previamente aplicamos "**clear**" o "@**nopass**" podemos restaurar las contraseñas globales con las correspondientes a continuación: "@_**site**_**edit**", "@_**site**_**read**" or "@_**site**_**upload**" (si este último está establecido).

Nosotros como administradores podemos...

1. Establecer contraseñas globales para páginas y/o grupo de páginas que no tengan contraseñas.
2. Usar **attr** para controlar quien es capaz de establecer contraseñas en las páginas.
3. Si está habilitada las subida de archivos, establecer contraseñas para ello.
4. Usar nuestra contraseña de administrador para eliminar o cambiar las contraseñas de lectura y/o escritura establecidas para cada página o grupo de páginas.
5. Usar "**SiteAdmin.AuthList**" para visualizar las páginas o grupos de páginas que tengan contraseñas establecidas.

Jerarquía de contraseñas.

- Las contraseñas de **páginas** privan sobre las contraseñas de **grupos de páginas**.
- Las contraseñas de **grupos de páginas** privan sobre las **contraseñas globales**.
- Las contraseñas de **administrador** privan sobre las **contraseñas globales, contraseñas de grupos de páginas y contraseñas de páginas.**

Tapando los agujeros de orden y seguridad.

- Por defecto cuando editamos una página es opcional el colocar nuestro nombres para indicar autoría, esto es especialmente útil si nuestra wiki está abierta a todo público, completamente libre. Pero si estamos en el punto de haber creado cuentas de usuario debemos obligar a que utilizen su nombre en cada edición, esto se logra en el archivo **config.php**:
    -

```
$Author = $AuthId; # after include_once()
```

- También podemos permitir al usuario escribir el nombre que desee *pero siempre aparecerá primero un prefijo que lo identifica como autor.* Esto es útil si el autor desea compartir su clave con otro autor que nosotros no hemos autorizado y se desea saber quien hizo qué. **En todo caso el autor/usuario con la cual se modificó la página o grupo de páginas será plenamente responsable de dichos cambios, sin importar si otra persona lo hizo.**
- Otra forma más estricta es inhabilitar el campo "autor" en el formulario de edición de cada página, pero eso cierra la posibilidad de dejar abiertas ciertas páginas o grupos de páginas (partiendo del principio de que cada quien desea ser conocido por los cambios realizados).
- Si tenemos varias wikis corriendo en un mismo servidor (diferentes dominios o subdominios) debemos decirle a PHP que cada wiki tiene su propia y particular sesión agregando al principio de **config.php** la siguiente línea **(por supuesto, generad vuestr propia clave aleatoria, y una por cada wiki que tengáis):**
    -

```
session_name('XYZSESSID');
```

- Podemos establecer varias contraseñas para 'admin' con array() en el archivo config.php recordando el habilitar cuentas de usuario y contraseñas, forzando el grabado de autorías (ver párrafos anteriores) para saber quién hizo qué.
- Podemos predefinir los grupos de páginas que contendrá nuestra PmWiki (más bien muchos verán esto como una restricción pero todo está en el diálogo entre los autores) por medio de insertar en **config.php** la siguiente clave: "**$GroupPattern = '(?:Sitio|PmWiki|Principal|Ventas|Compras)';**" y si necesitamos ampliarla pues le agregamos otro caracter tubería seguido de los nuevos grupos.

Manejando la "internacionalizaciones".



Las llamadas "internacionalizaciones" ("**i18s**"por su abreviatura en inglés de las 18 letras -22 en castellano-) son las traducciones a otros idiomas de **PmWiki**. Como bien sabemos el lenguaje **PHP** fue escrito en idioma inglés y **PmWiki** no es la excepción. La solución que planteó el Dr. Patrick Michaud está plasmada en dos aspectos fundamentales: la traducción de los "diálogos comunes" y la creación por parte de la comunidad de las respectivas páginas "tutoriales" que describen la instalación, uso y mantenimiento de **PmWiki**. Por ello divideremos esta sección en dos temas "apartes".

Traducción de los "diálogos comunes".

En **PmWiki** "normalizaron" los diálogos comunes, que son las instrucciones que visualiza el usuario. Las más visibles de ellas son "*View Edit History Print Backlinks*" que vemos en la parte superior derecha de cada página. El mecanismo es el siguiente: existe un archivo principal, el núcleo o corazón de **PmWiki** llamado "*pmwiki.php*" el cual nuestro servidor web ejecuta y carga los diversos componentes -que no vienen al caso por la complejidad del asunto-.

Lo que queremos *ejemplarizar* es cuando se carga el archivo "**Site.PageActions"** en la carpeta **wililib.d** allí tenemos unos cuantos mensajes al usuario, uno de ellos un enlace para modificar la página que esté en ese momento visualizando, otro para ver su historial además de las pa?inas relacionadas (y hay más enlaces/opciones).

En forma de matriz llamada **text()** se cargan los diferentes mensajes predefinidos por una estructura (de nuevo, estamos simplifanco al máximo) hasta llegar a los simples mensajes *que acompañan una "acción", cargar **pmwiki.php** más el parámetro a ejecutar (cuando el usuario hace click en la opción/enlace correspondiente):*

- "Edit" -> "pmwiki.php?action=edit"
- "History" -> "pmwiki.php?action=diff"
- "Backlinks" -> "pmwiki.php?action=search&q=link={*$FullName}" (donde $FullName es una variable: el nombre de la página que estamos visualizando y sus páginas *relacionadas **-no existen jerarquías en PmWiki, solo existe en nuestros cerebros-***)

Aquí es que vamos a llegar al meollo del asunto: dichas palabras "Edit", "History" y "Backlinks" aparecen en el código *no como texto en sí, sino como "variables":* **"$[Edit]", "$[History]" y "$[Backlinks]"** de manera tal que esas variables definidas se puedan especificar en un archivo con la extensión "**.XLPage**" con el siguiente contenido según nuestros ejemplos seleccionados:

- 'Edit' => 'Editar'
- 'History' => 'Historial'
- 'Backlinks' => 'Retroenlaces'

Resulta ser que para nosotros ***en supuesto "idioma español"*** nos asignaron dos letras "Es" de esta manera nuestro idioma vienen ser definidos los mensajes comunes en un archivo llamado "**PmWiki**Es**.XLPage**". Pensamos que deberíamos diferenciarnos porque en España se hablan varioas idiomas: gallego, castellano, catalán, vasco y portugués... ***que está bien, Portugal se independizó y cogió otro rumbo y otro idioma pero forma parte de lo que los romanos al civilizar llamaron "Hipania" (perdón por la lección de historia, je, je, je). Somos de la idea de***

que deberíamos tener OTRAS letras como, se nos ocurre "cas" (ya "ca" está tomado por el idioma catalán) pero en fin, todos llevamos dentro nuestro pequeño "Don Quijote de la Mancha" y en cada molino vemos fieros gigantes ¡ja!

Traducción de las páginas en sí.

Al abrir la página principal de **PmWiki** *sin haberla internacionalizado (a eso vamos)* en el panel izquierdo ("slide panel") aparecerá varias opciones -en inglés- de las cuales las tres primeras (hay site opciones en esa sección) dirán:

- "Initial Setup Tasks"
- "Basic Editing"
- "Documentation Index".

Si colocamos el puntero sobre ellas, sin hacer click, veremos que sus *enlaces web* (llamados a **pmwiki.php** acompañados de un parámetro) serán los siguientes -coloreamos para que veáis las diferencias, *no colocamos el enlace completo porque cada dominio web es diferente*-:

- pmwiki.php?n=PmWiki.InitialSetupTasks
- pmwiki.php?n=PmWiki.BasicEditing
- pmwiki.php?n=PmWiki.DocumentationIndex

Lo coloreado en color marrón son "páginas del sistema" de **PmWiki** y son las que debemos traducir ¿recordáis las dos letras de código de nuestro idioma? Pues las debemos intercalar para que apunten a los archivos que vamos a descargar con las traducciones:

- pmwiki.php?n=PmWikiEs.InitialSetupTasks
- pmwiki.php?n=PmWikiEs.BasicEditing
- pmwiki.php?n=PmWikiEs.DocumentationIndex

Del siguiente enlace podréis descargar los idiomas disponibles en incluso un "megapaquete" con todas las traducciones. Nosotros solo descargaremos el archivo "**i18n-es.zip**" el cual descomprimiremos y "subiremos" a nuestro servidor **teniendo cuidado de darle los permisos de lectura necesarios** y los ubicaremos todos en la carpeta **wikilib.d**

Activando el idioma deseado.

En este punto debemos editar de nuevo el dichoso fichero llamado "**config.php**" que está ubicado en la carpeta **/local** y modificar y adicionar las siguientes líneas:

```
include_once("scripts/xlpage-
utf-8.php"); # optional  XLPage('es','PmWikiEs.XLPage');
```

La primera línea en realidad la *descomentamos* porque viene así de la plantilla por defecto, y ese comentario que indica "optional" osea opcional *para nosotros es obligatorio (ya eso lo explicamos en* nuestro tutorial HTML)*.* La segunda línea indica el llamar a la función **XLPage** y tiene dos parámetros:

- El primer parámetro son las letras que mostrará en el cintillo inferior *en el caso que instalemos varios idiomas,* ***no tiene nada que ver con la codificación ISO de dos letras***. Por razones estéticas y debido a la cantidad de idiomas y que ninguno destque de los demás se escoge ese formato *pero nada nos impide colocarle "español" o "castellano"; queda a vuestro criterio -si manejais pocos idiomas-.*
- El segundo parámetro es el nombre exacto del archivo porque los servidores GNU/Linux distinguen mayúsculas de minúsculas. Aquí queda de parte de nosotros mantener la clave ISO "Es" para claridad de nuestros archivos, debemos ser conscuentes en esto porque nuestros grupos de páginas llevarán siempre el perfijo **PmWikiEs.**

Mejorando la selección de idioma.

No sabemos quien visitará nuestra página en este mundo tan globalizado, pero si utilizan un navegador web que se haya programado según las normas del W3C lo mejor que podemos hacer es incluir en **config.php** el siguiente guion:

[cc lang="php" tab_size="2" lines="40"]$lang = substr($_SERVER['HTTP_ACCEPT_LANGUAGE'], 0, 2);
switch ($lang){
case "es":
XLPage('es','PmWikiEs.XLPage');
break;
case "en":
XLPage('en','PmWikiEn.XLPage');
break;
default:
XLPage('en','PmWikiEn.XLPage');
break;
}[/cc]

En la primera línea por medio de PHP le preguntamos al servidor web el encabezado con que solicita la página nuestro visitante y extraemos las dos primeras letras para luego seleccionar de los idiomas disponibles que tengamos instalados, y por defecto, si no se consigue, pues lo mostramos en ingéls, el lenguaje "materno" de **PmWiki**. Lo hemos probado con Mozilla Firefox 53 y funciona.

Actualizado el sábado 5 de mayo de 2018

PMWiki funciona como la Wikipedia, está abierta para que todos y todas coloquemos nuestras sugerencias y propuestas. Por ello el 13 de junio de 2017 propusimos en la página oficial del Proyecto PMWiki el código anterior y a la fecha de hoy la propuesta se mantiene, *ergo,* no la han eliminado y suponemos que ya con tanto tiempo transcurrido queda aprobada de manera implícita. Dicha propuesta la pueden consultar en este enlace y si aun dudaís de nuestra autoría basta conq ue hagáis click en el historial del artículo. Fue un honor haber contribuido en algo al tema.

Visualizando por primera vez nuestra página en nuestro idioma.

Pues tan simple como recargar la página y los mensajes comunes estarán en nuestro idioma... ***pero hasta allí llega nuestra "alegría de tísico" (por lo poco que dura antes de entrar en un atque de tos).*** Resulta que igual nos aparece en inglés todo lo demás y he aquí que debemos comenzar a trabajar.

La primera página que modificaremos es precisamente la principal llamada "**Main.HomePage**" la cual, por ahora, no hemos encontrado manera de cambiar al castellano. Lo que es cierto es que podemos -y debemos- modificarla borrando todo el contenido en inglés e ir creando nuestro propio contenido para luego modificar el panel izquierdo ("edit SideBar").

Nuestro primer grupo de páginas.

Aunque suene extraño vamos a aventurarnos a crear dos grupos de páginas de una vez. Para ello ya estamos modificando la "**Main.HomePage**" (luego de haber introducido nuestra contraseñade edición, si la hubiéramos establecido) y escibiremos el siguiente código:

```
[[WikiAeroplanos | Aviones]]  * [[WikiAeroplanos.Cazas | Uso militar]]  *
 [[WikiAeroplanos.Pasajeros | Transporte]]  * [[WikiAeroplanos.Hidroavion
es | Mixto]]    [[WikiTrenes | Trenes]]  * [[WikiTrenes.Electricos | Sigl
o XXI]]  * [[WikiTrenes.Vapor | Siglo XX]]
```

Le damos guardar y al recargar la ápgina veremos lo siguiente:

¿Observáis los signos de interrogación *que nosotros no escribimos?* **Pues es la maner de PmWiki indicarnos que dicho tema no ha sido escrito**. Aquí radica la sencillez y rapidez de **PmWiki**: cuando hacemos click se abre inmediatamente para editar cada una de las págins que queremos desarrollar. He aquí que le vamos agregar un poco de complejidad a cada una de ellas.

Fuentes consultadas.

En idioma francés.

- «Sortie de PmWiki 2.2.29» modéré par Lucas Bonnet.

En idioma inglés.

- «Basic PmWiki editing rules».
- «PmWiki + SQLite».
- «Working Together With Wikis» by Anil Emrajani.
- «Setting language preferences in a browser» W3C Internationalization.
- «Detect Browser Language in PHP: Why dont you keep it simple and clean?».