

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN



"Integración segura de WordPress con
NextCloud"

TRABAJO FIN DE GRADO

Septiembre – 2020

AUTOR: Nieves Marina Martínez Pujalte

DIRECTOR: Miguel Martínez Rach

DIRECTOR: Otoniel López Granado

DIRECTOR EXTERNO: Eduardo Urrea Alcaraz

Agradecimientos

En primer lugar, quiero agradecer el apoyo de mi familia por los ánimos que me han dado sin ellos no podría haber realizado esto. En especial a mi pareja Antonio por animarme a no abandonar y acabar la carrera después de unos años.

Agradecer a los profesores de la escuela quienes me han ayudado y enseñado durante estos años. Y en especial a Miguel y Otoniel quienes me han ayudado y aconsejado con el proyecto teniendo una gran paciencia y comprensión con el poco tiempo que disponía por el trabajo.



Nieves Marina Martínez Pujalte.

Resumen

Debido a la continua evolución de Internet, las empresas se han planteado un cambio en su modelo de negocio. Cada día es menos habitual relacionarse con una empresa de forma presencial. Actualmente los clientes interactúan con las empresas y viceversa a través de internet desde cualquier lugar y con cualquier dispositivo ya sea una Tablet, un portátil o móvil. Para el cliente es mucho más cómodo y las empresas reducen en costos. Este cambio en las empresas es lo que se conoce con el término de “transformación digital”. Sobre todo, las pequeñas empresas han incorporado este nuevo modelo de negocio. El objetivo de estas es ser mucho más competitivas, aunque conlleve algún impedimento.

Puesto que las empresas han tropezado con un inconveniente, la volumetría de los datos. La cantidad de datos que recogen de los dispositivos es bastante grande y necesitan guardarlos. Por ello no se pueden conformar con cualquier almacenamiento web, sino que necesitan de un almacenamiento virtual. Este término se conoce como nube o “cloud computing”. Uno de los sectores que más han apostado por este servicio es el sector de la abogacía. Los bufetes de abogados usan este servicio para acceder a sus agendas, gestores de correo, intercambiar plantillas o archivos confidenciales con los clientes de forma rápida desde cualquier lugar o con cualquier dispositivo.

Para el intercambio de archivos podrían utilizar cualquier gestor de archivos. Algunos de los más conocidos son Dropbox, Google Drive, Mega o similares. Aunque estos tienen su propio dueño y tendrá acceso a los archivos “privados” que se almacenan en él. Para los bufetes de abogados sería ideal ser los únicos propietarios del gestor de archivos. La única forma de conseguirlo es teniendo los archivos en su propio servidor. De esta forma se seguiría conservando la privacidad.

En este proyecto se dotará de un software de cloud al abogado para que mediante una instalación pueda tener un gestor de almacenamiento de archivos en su propio servidor de manera “segura”. Este software será un plugin para uno de los gestores de páginas web más utilizados en estos últimos años, WordPress. El abogado podrá intercambiar archivos de una forma mucho más eficaz y cómoda con sus clientes desde cualquier lugar o dispositivo instalando este software siempre teniendo contratado un proveedor de hosting.

ÍNDICE

Agradecimientos.....	2
Resumen.....	3
Índice de Imágenes.....	6
Índice de tablas.....	8
Capítulo 1: Introducción.....	9
1.1 Introducción.....	10
1.2 Objetivos.....	12
1.3 Organización de la memoria.....	13
Capítulo 2: Cloud Computing.....	14
2.1 Historia.....	15
2.2 Cloud Computing.....	17
2.3 Estado del Arte.....	21
2.4 Herramientas de desarrollo.....	22
Capítulo 3: Herramientas.....	23
3.1 NextCloud.....	24
3.2 WordPress.....	27
3.3 Instalación WordPress y NextCloud.....	30
Capítulo 4: Aspectos de Seguridad.....	39
4.1 Seguridad.....	40
4.2 Seguridad WordPress.....	41
4.3 Seguridad NextCloud.....	45
Capítulo 5: Solución Integrada.....	48
5.1 WordPress Plugin.....	49
5.2 NextCloud mediante API.....	53
Capítulo 6: Plugin WPCloud.....	55
6.1 PLUGIN WPCLOUD.....	56
6.2 ACCESO “MI CLOUD”.....	56
6.3 ALTA USUARIOS.....	58
6.4 VISUALIZACIÓN DE CARPETAS.....	61
6.5 HERRAMIENTAS.....	65
6.5.1 Crear carpeta.....	66

6.5.2 Crear fichero	67
6.5.3 Subir fichero.....	68
6.5.4 Borrar fichero	71
6.5.5 Descargar archivos	72
6.6 GESTIÓN USUARIO.....	74
Capítulo 7: Conclusiones y Líneas Futuras	76
7.1 Líneas futuras	77
7.2 Conclusiones.....	78
7.3 Presupuesto.....	78
Bibliografía.....	79
Bibliografía de imágenes.....	83



Índice de Imágenes

[Figura 1: Dispositivos conectados a la nube]	10
[Figura 2: Beneficios Startups]	11
[Figura 3: Informáticos John McCarthy y Licklider]	15
[Figura 4: Logos proyectos en la nube]	16
[Figura 5: Tipos básico servidores].....	17
[Figura 6: Servicios cloud computing SaaS, PaaS y IaaS]	19
[Figura 7: Tipos de nube según recursos]	20
[Figura 8: Cloud Privados].....	21
[Figura 9: Icono Sublime]	22
[Figura 10: Icono Bitwise SSH Client]	23
[Figura 11: Gestores de archivos con código abierto]	24
[Figura 12: Menú administración WordPress].....	29
[Figura 13: Barra herramientas y zona publicación de WordPress].....	30
[Figura 14: Página web WordPress].....	30
[Figura 15: Página web XAMPP].....	31
[Figura 16: Instalación XAMPP paso 1].....	31
[Figura 17: Instalación XAMPP paso 2].....	32
[Figura 18: Instalación XAMPP paso 3].....	33
[Figura 19: Instalación XAMPP paso 4].....	33
[Figura 19: Instalación WordPress paso 1].....	34
[Figura 20: Instalación WordPress paso 2].....	34
[Figura 21: Instalación WordPress paso 3].....	35
[Figura 22: Instalación WordPress paso 4].....	35
[Figura 23: Instalación WordPress paso 5].....	36
[Figura 24: Instalación WordPress paso 6].....	36
[Figura 25: Instalación NextCloud paso 1].....	37
[Figura 26: Instalación NextCloud paso 2].....	37
[Figura 27: Instalación NextCloud paso 3].....	38
[Figura 28: Instalación NextCloud paso 4].....	38
[Figura 29: Código seguridad WordPress paso 1].....	43
[Figura 30: Código seguridad WordPress paso 2].....	43
[Figura 31: Código seguridad WordPress paso 3].....	44
[Figura 32: Código seguridad WordPress paso 4].....	44
[Figura 33: Código seguridad WordPress paso 4].....	44
[Figura 34: Seguridad NextCloud – historial de cambios].....	45
[Figura 35: Seguridad NextCloud – integración aplicaciones de redes]	45
[Figura 36: Seguridad NextCloud – información para auditorías].....	46
[Figura 37: Seguridad NextCloud – gestión archivos aplicación Flow]	46
[Figura 38: Seguridad NextCloud – protocolo “Secure Sockets Layer”]	47
[Figura 39: Instalación Plugin IncludeMe].....	49

[Figura 40: Instalación plugin paso 1]	50
[Figura 41: Instalación plugin paso 2]	50
[Figura 42: Instalación plugin paso 3]	51
[Figura 43: Instalación plugin paso 4]	51
[Figura 44: Menú WPcloudPlugin]	52
[Figura 45: Código librerías]	52
[Figura 46: Logo Sable/dav]	53
[Figura 47: Instalación Sable/dav paso 1]	53
[Figura 48: Instalación Sable/dav paso 2]	53
[Figura 49: Conexión cliente WebDav de NextcCloud]	53
[Figura 50: Listado clientes NextCloud]	54
[Figura 51: Propiedades usuarios]	54
[Figura 52: Menú Mi Cloud paso 1]	56
[Figura 53: Página Mi Cloud paso 2]	56
[Figura 54: Acceso MiCloud paso 3]	56
[Figura 55: Implementación acceso MiCloud paso 4]	57
[Figura 56: Implementación acceso MiCloud paso 5]	57
[Figura 57: Menú Alta Usuarios paso 1]	58
[Figura 58: Formulario Alta Usuarios paso 2]	58
[Figura 59: Carpeta usuario paso 3]	59
[Figura 60: Creación usuario WordPress]	59
[Figura 61: Crear usuario NextCloud]	60
[Figura 62: Crear carpeta cliente]	60
[Figura 63: Compartir carpeta]	61
[Figura 64: Visualización carpetas paso 1]	61
[Figura 65: Visualización carpetas paso 2]	62
[Figura 66: Visualización carpetas paso 3]	62
[Figura 67: Cute File Browser paso 4]	62
[Figura 68: Array especifico paso 5]	63
[Figura 69: Visualización carpetas paso 6]	63
[Figura 70: Visualización carpetas paso 7]	63
[Figura 71: Array directorio paso 8]	64
[Figura 72: Barra herramientas]	65
[Figura 73: Menú ratón]	65
[Figura 74: Botón crear carpeta paso 1]	66
[Figura 75: Popup crear nueva carpeta paso 2]	66
[Figura 76: Creación nueva carpeta paso 3]	66
[Figura 77: Creación nueva carpeta paso 4]	66
[Figura 78: Creación nuevo fichero paso 1]	67
[Figura 79: Popup nuevo fichero paso 2]	67
[Figura 80: Creación nuevo fichero paso 3]	67
[Figura 81: Menú subir fichero paso 1]	68
[Figura 82: Menú subir fichero 2]	68
[Figura 83: Menú subir fichero 3]	68
[Figura 84: Subir fichero paso 4]	69
[Figura 85: Carpeta temporal paso 5]	69
[Figura 86: Subir archivo paso 6]	70

[Figura 87: Subir archivo paso 7].....	70
[Figura 88: Menú borrar paso 1]	71
[Figura 89: Popup borrar paso 2].....	71
[Figura 90: Borrar fichero paso 3].....	71
[Figura 91: Menú descargar archivo paso 1]	72
[Figura 92: Descargar archivo paso 2]	72
[Figura 93: Descargar archivo paso 3]	72
[Figura 94: Descargar archivo paso 4]	73
[Figura 95: Descargar archivo paso 5]	73
[Figura 96: Menú gestión usuario paso 1]	74
[Figura 97: Gestión usuario paso 2]	74
[Figura 98: Gestión usuario paso 3]	74
[Figura 99: Gestión usuario paso 4]	75
[Figura 100: Gestión usuario paso 5]	75
[Figura 101: Gestión usuario paso 6]	75

Índice de tablas

[Tabla 1: Licencias NextCloud].....	26
[Tabla 2: Comparativa gestores de contenido].....	27



Capítulo 1: Introducción

En este primer capítulo se explicará con una breve descripción de la evolución de la tecnología de Internet. Llegando a necesitar un sistema de almacenamiento y ejecución de la nube a través de la red. Además, explicaremos cuál es el objetivo de nuestro proyecto y las partes en las que se ha dividido esta memoria.

1.1 Introducción

En la actualidad, la evolución de la tecnología de Internet está en continuo auge. El uso continuado del Internet está siendo indispensable en la población tanto en la vida social como en la laboral, teniendo la necesidad de conectarse sea cual sea el momento y/o lugar. Cada día tenemos más dispositivos y/o sensores conectados a internet. Inclusive hemos conectado objetos cotidianos como alarmas, televisores, ollas, aspiradores, relojes o incluso ropa textil a la red de Internet. Esta tecnología popularmente se le conoce como “el Internet de las cosas” (“Internet of Things”).



[Figura 1: Dispositivos conectados a la nube]

Debido a la gran cantidad de dispositivos que tenemos conectados, y cada día van en aumento, nos surge la necesidad de poder guardar los datos necesarios en nuestros ordenadores, tablets, móviles, sensores, etc. Aunque muchas veces debido al volumen de los datos nos es imposible almacenarlos en estos dispositivos. Necesitaremos aplicar un servicio de la Tecnología IoT (“Internet of Things”) que se denomina “Cloud Computing” o “nube”, el cual nos ayudará a almacenar datos o programas en los servidores, accediendo posteriormente a ellos a través de Internet.

Esto hará que nos ahorremos dinero, pudiendo ejecutar aplicaciones de alta potencia sin la obligación de tener un ordenador personal con altos requerimientos computacionales. Es decir, podemos permitirnos un ordenador con un disco duro más pequeño y un procesador menos caro y potente, ya que las aplicaciones no las deberemos tener instaladas en nuestro ordenador, debido a que éstas están basadas en web. Los gastos serán más bajos porque

pagaremos por “servicios” o por gigabytes de almacenamiento sin la necesidad de ir actualizando nuestros equipos tan frecuentemente.

No obstante, la gran ventaja que tiene la nube es facilitarnos poder acceder desde cualquier dispositivo o lugar a nuestros datos o servicios, siempre que tengamos acceso a Internet, pudiendo entrar a cualquier red social, enviar un correo o encender cualquier electrodoméstico sin estar físicamente en casa.

Sin embargo, donde más se ha visto implementada esta tecnología no es en la vida cotidiana sino en la empresarial. Como describimos anteriormente el avance que ha sufrido la tecnología ha hecho replantearse a las empresas un cambio en sus modelos de negocio, sufriendo lo que se conoce como “Transformación Digital” pudiendo tener más oportunidades de negocio y siendo mucho más competitivas, sobre todo en las pymes o pequeñas empresas emergentes (también conocidas como “startups”).



[Figura 2: Beneficios Startups]

Éstas pueden rivalizar con grandes empresas ya que, debido a tener un bajo coste, no es necesario invertir gran cantidad de dinero en publicidad, infraestructuras, almacenamiento de datos, licencias de software o hardware. Todo tipo de sectores empresariales apuestan por gestionar sus negocios mediante la nube como, por ejemplo: cadenas de hoteles, abogados, sector salud y agricultura entre otros muchos.

1.2 Objetivos

El principal objetivo de este proyecto es la puesta en práctica de los conocimientos y destrezas aprendidos en la carrera de Grado de Ingeniería en Tecnologías de Telecomunicación, aplicados a un problema real mediante una investigación y un planteamiento de una solución basada en la nube para el ámbito de la abogacía.

Como hemos indicado en el apartado anterior la irrupción de Internet ha provocado que las pequeñas empresas apuesten por una transformación digital para ser mucho más competitivas. Uno de los sectores que se ha comprometido a cambiar su modelo de negocio es el sector de la abogacía. Para que una empresa de un pequeño abogado pueda competir contra grandes bufetes, la única solución es reducir costes. Para conseguirlo, la nube les permitirá reducir los costes sin prescindir del último software o hardware, almacenamiento e intercambio ágil de archivos con los clientes.

Sin embargo, uno de los grandes problemas que surgen, es el referente a la seguridad de la información y la confidencialidad. Los abogados guardan en la nube una serie de archivos que deben preservar la confidencialidad y estar protegidos. Sin embargo, si usan un gestor de archivos como Dropbox, Google Drive, Box, Mega o similares, sus propietarios pueden llegar a tener información privilegiada de los clientes y podrían llegar a usarse en su contra.

Surge entonces la necesidad de poder tener un gestor de archivos en el propio servidor de la empresa. De esta manera el buffet de abogados sería “el propietario” y no tendría acceso a los archivos alguien externo a la empresa. Este gestor de archivos debería poder integrarse fácilmente en la página web de los bufetes para que los clientes puedan acceder de forma sencilla.

La mayoría de las empresas, tanto grandes como Pymes, utilizan WordPress para explotar su página web. WordPress es un gestor de contenidos para no tener que realizar páginas web desde cero. Sus ventajas principales son: es una plataforma de código abierto, fácil de usar, personalizable, facilidad de adaptación a móviles, está en continuo desarrollo para no quedarse obsoleto y fácil gestión de usuarios.

Nuestro objetivo principal en este proyecto será dotar de un software de cloud o nube a los bufetes de abogados. Para que, mediante la instalación de un plugin en WordPress, éste pueda tener un gestor de almacenamiento de archivos en su propio servidor de una manera “segura” y que le permita intercambiar archivos de una forma mucho más eficaz y cómoda con sus clientes desde cualquier lugar.

1.3 Organización de la memoria

La memoria consta de 7 capítulos, de los cuales a continuación se hará una breve descripción:

- Capítulo 1 Introducción: se realizará una pequeña explicación sobre la evolución que han sufrido las empresas mediante la transformación digital. Además, cómo ha influido “la tecnología del internet de las cosas” y el servicio “cloud” en las pequeñas y medianas empresas. Se describirán brevemente los objetivos de este proyecto y cómo está organizada esta memoria.
- Capítulo 2 Cloud Computing: se repasará la historia de la computación en la nube y algunos de los proyectos que han influido en la evolución de la nube. Además, se hará una pequeña explicación para saber en qué consiste la computación en la nube. Se añadirán algunos programas que necesitaremos instalar en nuestro ordenador para la realización de nuestro proyecto.
- Capítulo 3 Herramientas: conoceremos algunos sistemas de archivos de código abierto, en especial el que vamos a usar, NextCloud. Seguidamente conoceremos un gestor de contenido más usado para realizar la página web, WordPress. Para acabar con el capítulo la instalación de ambas.
- Capítulo 4 Aspectos de seguridad: en este capítulo estudiaremos brevemente la seguridad en sistemas cloud, en WordPress y en NextCloud para realizar un plugin “más seguro”.
- Capítulo 5 Solución Integrada: en este apartado explicaremos dos herramientas básicas para desarrollar el plugin WPCloud. Se explicará cómo desarrollar un plugin dentro del gestor de contenido de WordPress y cómo usar la API de NextCloud para que se puedan comunicar.
- Capítulo 6 Plugin WPCloud: conoceremos el funcionamiento del plugin que hemos desarrollado.
- Capítulo 7 Conclusiones y líneas futuras: finalmente se analizará algunas mejoras a futuro, el presupuesto y unas conclusiones.

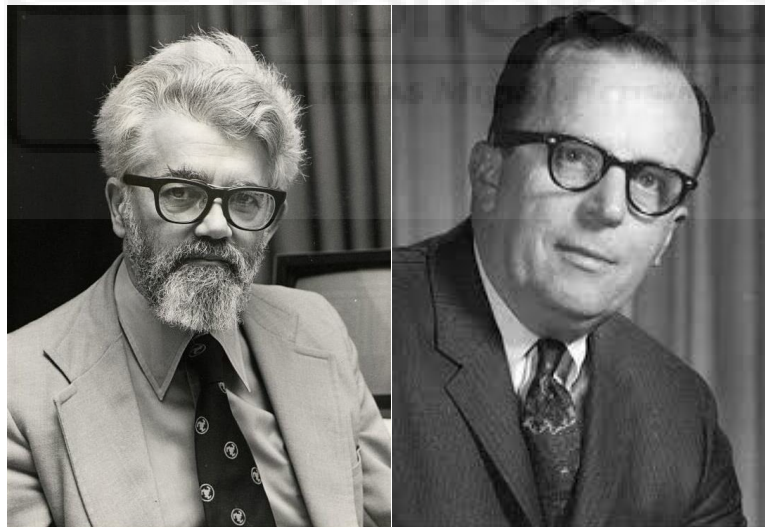
Capítulo 2: Cloud Computing

En este segundo capítulo se repasará la historia del término Cloud Computing y de las tecnologías IoT. Se estudiará la descripción de la computación en la nube y finalmente compararemos cómo las tecnologías IoT han influido en estos últimos años.

2.1 Historia

El término “Cloud Computing o computación en la nube” se remonta a los años 60 sobre 1961. El informático John McCarthy propuso compartir recursos entre empresas para ahorrar en gastos con el objetivo de que venderse como un servicio de luz o agua. Algunas empresas comenzaron a compartir infraestructuras, software y hardware. Como, por ejemplo, compartían oficina por horas, entornos de desarrollos, programas informáticos, pagaban por kilobytes almacenados mensualmente, horas de uso de software entre otras muchas opciones.

Pero no llegó a cuajar. Incluso aunque el informático Licklider volviera a introducir la idea de computación compartida sobre 1969. El uso de esta idea fue disminuyendo debido a que el ancho de banda todavía no estaba preparado para soportar el peso de la nube.



[Figura 3: Informáticos John McCarthy y Licklider]

Sin embargo, a finales de 1999 Salesforce.com ofreció aplicaciones empresariales a través de una página web, y a partir de ese momento las empresas comenzaron a interesarse por la nube. En ese mismo año en una presentación sobre el estudio de radiofrecuencias (RFID – “Radio Frequency Identification”) y sensores emergentes en el Instituto de Massachusetts (MIT) se comenzó a hablar sobre el término de “tecnología IoT (“Internet of Things”, la tecnología del internet de las cosas).

No obstante, este término emergió entre el año 2008 y 2009 donde el número de personas que tenían un dispositivo conectado a Internet fue en aumento hasta el día de hoy. Esto permitió desarrollar el protocolo denominado IP (“Internet Protocol”, protocolo de internet) permitiendo transportar paquetes de datos sin asegurar la recepción de estos.

Aunque años atrás el primer dispositivo que utilizó la misma comunicación como si se conectara un dispositivo a la nube fue una máquina de Coca Cola en la Universidad de Carnegie Mellon por los años 80. Esta utilizaba un protocolo M2M (“machine to machine”) comunicación máquina a máquina.

También podríamos decir que “la tecnología IoT” (“Internet of Things”) está basada en un sistema SCADA (“Supervisory Control And Data Acquisition”). Este sistema necesita de un software para controlar y adquirir datos.

En 2007 gracias a la colaboración de IBM y Google en varias universidades como la de Massachussets se comenzó a introducir el concepto de computación en la nube. IBM aceleró las infraestructuras. Mientras que Google creó la aplicación Apps Engine para programadores, servicios de seguridad y archivado de correo electrónico. Esto aceleró el estudio de clouds privados.



[Figura 4: Logos proyectos en la nube]

A principios del siguiente año, se abarcó un proyecto denominado Eucalyptus. Fue la primera plataforma de código abierto compatible con la API (“Application Programming Interfaces”) de AWS (“Amazon Web Service”). El conjunto de definiciones y protocolos que se desarrollaron para crear e integrar el software de AWS se utilizó para el despliegue de clouds privados.

Seguidamente en 2008, aunque varios años antes comenzará la investigación, surgió el proyecto que actualmente se conoce como Open Nebula System. En él se crearon soluciones para gestionar máquinas virtuales en infraestructuras

distribuidas. En 2019 Microsoft lanzó Windows Azure. Y Google lanzó sus propias aplicaciones, las cuales hoy se ejecutan en su nube.

Debido a todos los desarrollos sobre cloud en 2010 se definieron las estructuras que veremos en el siguiente apartado de esta memoria. A partir de ahí se lanzaron varias aplicaciones de hasta el día de hoy. Uno de los primeros fue Apple con su servicio iCloud con almacenamiento en la nube.

2.2 Cloud Computing

En la actualidad las empresas necesitan una página web para ser más visibles en el mercado y poder dar a conocer sus productos entre los clientes. Para ello deberemos contratar un dominio y un alojamiento en un servidor. El dominio será el “nombre” que tendrá nuestro sitio web y con el cual podrán acceder insertándolo en el navegador web. Para guardar los códigos y archivos necesarios para la creación de una página web necesitaremos un espacio en un servidor.

Hace unos años, las empresas tenían sus propios servidores. Los servidores son equipos programados específicamente para responder información a otros dispositivos como ordenadores, móviles, tablets entre otros. Normalmente las pequeñas o medianas empresas la posibilidad de tener su propio servidor es complicada. No suelen tener a una persona con los conocimientos necesarios para gestionar un servidor ni un presupuesto para abarcarlo. En vez de contratar a una persona casi exclusivamente para ello contratan un servicio a una empresa especializada, lo que se conoce como empresa de “hosting”. Las cuales proporcionan un espacio en su servidor y además nos pueden ofrecer otros servicios. Por ejemplo, aplicaciones para instalar o crear una web más fácilmente. Las empresas de hosting suelen ofrecer almacenamiento de varios tipos:



[Figura 5: Tipos básico servidores]

- **Compartido:** se comparte servidor con otros clientes. Suele ser el más usado sobre todo para empresas que se inician. Económicamente es el más asequible y suelen tener un panel de control para facilitar la usabilidad. Dependiendo del tráfico que tengan los otros clientes puede ralentizar tu página web. Se tiene poco control del servidor ya que la instalación y el mantenimiento lo realiza la empresa de hosting.

- **Virtual:** conocido con las siglas VPS (Virtual Private Server). Se comparte servidor con otros clientes, aunque limitado a una cantidad de memoria fija. De esta forma si otros clientes pueden tener tráfico de datos no afecta en tu página web. El coste es más caro que el compartido, aunque se podrá tener más control sobre el servidor y se deberá tener conocimientos para hacer el mantenimiento.
- **Dedicado:** económicamente es el más caro. Es parecido a tener tu propio servidor. Te asignan un servidor exclusivamente para tu sitio web. El control sobre el servidor es total y te proporcionan asesoramiento, aunque también se deberá tener conocimientos sobre los servidores. Al no compartir recursos con nadie, la seguridad es mayor.

Todos estos tipos de servidor son físicos, es decir, nuestra página web depende de una máquina. Si esta dejara de funcionar por cualquier motivo no podríamos acceder a ella. Para solucionarlo ha surgido la idea que se conoce como “nube o cloud”.

Al contratar este tipo de servicio la empresa de hosting realizará una copia de código de la página web en cada uno de sus servidores. Comprobarán si algún servidor tiene problemas. En ese caso, se redirigirá la página web a otro servidor. Al tener una copia facilitará la restauración para no perder información, si fuera necesario. Al ser un tipo de servidor compartido con otros clientes, como hemos indicado anteriormente, tenemos la posibilidad de que nuestra página vaya ralentizada dependiendo del tráfico de datos que tengan los demás. Por ello, la empresa de hosting gestionará el redireccionamiento de la web a servidores que estén libres o que tengan poco tránsito de peticiones.

El término de “nube” se ha integrado tanto para servidores como para otros servicios de red como almacenamiento y aplicaciones. Actualmente es lo que se conoce como “cloud computing o computación en la nube”.

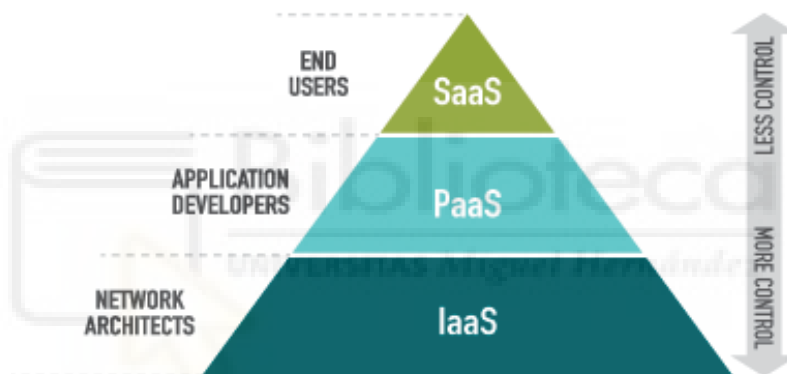
Según la Real Academia Española (RAE) se define como” el *modelo de prestación de servicios tecnológicos que permite el acceso bajo demanda a través de internet a un conjunto de recursos compartidos y configurables de modo escalable (como redes, servidores, capacidad de almacenamiento, aplicaciones y servicios) que pueden ser rápidamente asignados y liberados con una mínima gestión por parte del proveedor de internet*”. Es decir, un usuario puede acceder a cualquier servicio tanto de software como de hardware desde cualquier lugar y dispositivo, ya sea móvil, ordenador o Tablet a través de la red de internet gracias al almacenamiento en la nube.

El “cloud computing” lo utilizan cada vez más empresas por sus ventajas. Su principal beneficio es la movilidad que nos ofrece. Los clientes no deberán estar en la oficina o trabajar con un dispositivo predeterminado porque necesiten un software o hardware específico. Podrán acceder en cualquier lugar y con cualquier dispositivo. Además, el cliente ahorrará en gastos. Por ejemplo, ahorraría en infraestructuras, mantenimiento o copias de archivos. También el usuario no tendría que pagar por las renovaciones de licencias ni actualizaciones

de software. Las cuales son bastante costosas e en algunas ocasiones inaccesibles para pequeñas empresas. Otra de las ventajas que la nube nos aporta es el almacenamiento. El cliente puede contratar almacenamiento ilimitado. Incluso aumentar o disminuir la capacidad de la memoria según su necesidad.

La computación en la nube tiene alguna desventaja en el ámbito de seguridad. El software, hardware y los datos están en el servidor de una empresa externa. Por tanto, no somos los únicos propietarios y pueden usar la información que tengan almacenadas en contra del cliente. Por supuesto, el cliente necesitará una conexión a la red de Internet. Si hubiera un corte de luz o no tuviera conexión con internet no podría acceder a las aplicaciones.

El proveedor ofrece la computación en la nube de varias formas ya sea como software, plataforma o como infraestructura. Estos se explican a continuación:



[Figura 6: Servicios cloud computing SaaS, PaaS y IaaS]

- **SaaS (Software as a Service):** el proveedor facilitará el software y las aplicaciones utilizando un navegador web. No tendrá que instalarlas en su propio ordenador y paga sólo por el uso. Accede a ellas mediante web o suscribiéndose a la web del proveedor. El cliente podrá construir su propia plataforma. Reducirá los costes de hardware, software y soporte técnico. Aunque el usuario tendrá que aceptar la política de privacidad y contrato que proponga el vendedor. El convenio decidirá si el cliente o el proveedor es el dueño de la aplicación. Ejemplo: Office360 se paga por uso, pero no se compra la propiedad.
- **PaaS (Platform as a Service):** el cliente instala el software mediante una plataforma con hardware, sistema operativo y APIs (“Application Programming Interfaces”) que puedan solicitar al proveedor para crear una aplicación. Conlleva más gastos que la anterior ya que tiene coste de

soporte y software adicional. La desventaja que teníamos de seguridad está más repartida entre el proveedor y el cliente. El primero garantizará una plataforma segura. El segundo asegurará que su aplicación es correcta, segura y con el lenguaje de programación o las condiciones que tenga con el proveedor. Esta opción es interesante si se quiere desarrollar un software propio. Por ejemplo: Google App Engine la cual se pueden desarrollar aplicaciones en Java o Python.

- **IaaS (Infraestructure as a Service):** el proveedor entrega al cliente un acceso a la infraestructura entera. Se contrata por la capacidad del microprocesador y el almacenamiento. Es la opción más costosa, ya que el proveedor administra la infraestructura y su seguridad. Ejemplo, *EC2 de Amazon* aplicación donde se alquila ordenadores virtuales para ejecutar tu propia aplicación.

La opción más indicada dependerá de nuestras necesidades. SaaS si queremos consumir recursos, PaaS si queremos desarrollar o IaaS si queremos migrar.

También existen tres modos de tener una nube según los recursos que comparta: pública, privada o híbrida las cuales explicaremos a continuación:



[Figura 7: Tipos de nube según recursos]

- **Pública:** varios clientes comparten la infraestructura con unas condiciones pactadas con el proveedor. Tiene mayor eficiencia en los recursos, aunque menores costos por compartir. Utilizan el mismo servidor y posiblemente el mismo espacio de disco.
- **Privada:** los recursos se ofrecen de forma exclusiva ofreciendo mayor seguridad y pudiendo tener control es total sobre el servicio que se contrata.
- **Híbrida:** combinación de los dos servicios anteriores, es un poco más caro que la pública pero más barato que la privada. Para ello los servicios exclusivos son compartidos con otros.

Sobre todo, el “cloud computing” lo usan pequeñas y medianas empresas. Debido a la reducción de costos tienen la posibilidad de ser mucho más

competitivas. Uno de los sectores que ha apostado por este servicio es la abogacía. La nube va a permitir reducir costos en la infraestructura. Además, agilizar trámites con sus clientes pudiendo visualizar, editar, descargar, compartir de forma “segura” archivos desde cualquier lugar. Para ello surgirá la necesidad de tener un sistema de almacenamiento de archivos en la nube.

2.3 Estado del Arte

Durante estos últimos años han aparecido muchos sistemas de almacenamiento de archivos en la nube, la mayoría privados. La funcionalidad de todos ellos es muy parecida. Los archivos se pueden crear, copiar, modificar, compartir, subir a otras plataformas o simplemente visualizarlos en cualquier dispositivo desde cualquier lugar. A parte de coincidir en las mismas funciones, todos los gestores tienen un panel para poder gestionar los usuarios y los permisos necesarios para compartir o no archivos. La mayor diferencia entre gestores dependerá de los sistemas de seguridad que tenga. En muchos casos, esto determinará su precio.

Normalmente las tarifas dependen del soporte técnico, sistemas de seguridad, el espacio de almacenamiento que ofrecen o alguna herramienta extra. Suelen tener una versión gratuita con limitaciones, otra de pago y otra especial para empresas con alguna herramienta extra para gestionar equipos de trabajo o recuperar archivos borrados. Algunos gestores de archivos privados más conocidos son:



[Figura 8: Cloud Privados]

- **Dropbox:** apareció en 2007 por dos alumnos Drew Houston y Arash Ferdowsi del instituto de Massachusetts. La compañía tiene tres tarifas: Basic (2 gigabytes gratuitos), Plus (2 Terabytes por 10\$/mes) o Professional (3 Terabytes 17\$/mes). La desventaja de Dropbox es tener poco almacenamiento en la versión gratuita.
- **Google Drive:** su lanzamiento fue en abril de 2012. Creado de la fusión entre un editor de texto denominado Writely y un editor de hojas de cálculo Google Spreadsheets. hoy en día es lo que conocemos como Google Docs. Es uno de los gestores más utilizados. Ya que, puedes utilizar hasta

15 GigaBytes gratuitamente. Tiene varios planes según la capacidad de memoria que se necesite. El precio es parecido al anterior (2 TeraBytes por 10\$/mes)

- **iCloud:** pertenece a Apple desde su salida en octubre de 2011. Sincroniza gratuitamente a todos los dispositivos que tengan el sistema operativo de la misma empresa.
- **Onedrive:** muy parecida a la anterior perteneciente a Windows desde su salida en agosto de 2007.
- **Box:** es un gestor de archivos especializado para empresas desde su lanzamiento en mayo de 2009. Destaca por su seguridad en cifrado de archivos. Actualmente tienen 4 tarifas: *Starter*, *Business*, *Business Plus* y *Enterprise*. Las tarifas oscilan desde los 5 euros hasta los 32 € al mes. Todas tienen almacenamiento ilimitado, pero con restricciones sobre usuarios y límites de carga de archivos.

Como hemos indicado en apartados anteriores, el objetivo del proyecto será desarrollar un plugin en WordPress para gestionar archivos. Actualmente existe un plugin denominado Plugin File Manager. La url para acceder es <https://es.wordpress.org/plugins/file-manager/>. Este plugin es también un gestor de archivos, pero no tiene una interfaz para el usuario como la tiene el nuestro. Es más parecido a un software para transferir archivos conocido como FTP (File Transfer Protocol). Este protocolo se explicará con más detenimiento en otros capítulos.

2.4 Herramientas de desarrollo

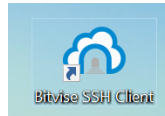
Para el desarrollo de nuestro plugin en WordPress necesitaremos el siguiente software para desarrollo:

- **Sublime Text:** editor de texto, se puede descargar del siguiente enlace: <https://www.sublimetext.com/3>



[Figura 9: Icono Sublime]

- **Bitvise SSH Client:** para acceder a la FTP de tu hosting. Podremos descargarlo del siguiente link : <https://www.bitvise.com/ssh-client-download>



[Figura 10: Icono Bitvise SSH Client]



Capítulo 3: Herramientas

En este tercer capítulo conoceremos las herramientas y conceptos que necesitaremos para el desarrollo del software. Estudiaremos los diferentes sistemas de archivos de código abierto que tiene el mercado. Conoceremos el motivo por el cual hemos elegido NextCloud para nuestro desarrollo. También tendremos una explicación del gestor de contenido más popular denominado WordPress. Finalmente, estudiaremos los pasos para instalar estas dos herramientas.

3.1 NextCloud

Los bufetes de abogados trabajan con una gran cantidad de documentos. Para ellos es conveniente usar un gestor de archivos en la nube. A cualquier abogado le ayudaría a organizarse mejor y a disminuir el tiempo de respuesta con el cliente. Sin embargo, no pueden elegir cualquier gestor de archivos. Tienen la necesidad de almacenar mucha información personal y los documentos deberán preservar su confidencialidad y ser protegidos. Por esta razón los bufetes dudan a la hora de subir archivos a la nube. Debido que el dueño del servidor donde tienen almacenados los archivos confidenciales es otra persona externa a la empresa y no el propio “bufete” y podrían utilizar la información confidencial en su contra.

Algunos de los gestores de archivos privados más conocidos son Drive, Dropbox, Mega u otras bastante conocidas. Una alternativa a ellos sería instalar una aplicación con en la cual podamos gestionar el almacenamiento de archivos, pero en nuestro propio servidor. Es decir, el propio bufete de abogados sería el único dueño del gestor de archivos. El tipo de software que necesitamos se conoce como “cloud open source” o servicio cloud de código abierto. Este software se puede instalar en cualquier servidor privado. Como su propio nombre indica es de código abierto, es decir, su código fuente está abierto. Algunos de los más conocidos en el mercado son Owncloud, NextCloud y Seafile. Los cuales detallaremos a continuación.



[Figura 11: Gestores de archivos con código abierto]

El software Owncloud tiene la versión más estable basada en el servidor web Apache. Utiliza bases de datos en *PHP* (“Hypertext Preprocessor”, lenguaje de programación para desarrollo web) y *MySQL* (“Structured Query Language”, sistema de gestión de bases de datos). Además, está escrito en lenguajes de programación *PHP* (“Hypertext Preprocessor”) y *JavaScript* (lenguaje de programación orientado a objetos). Este software está disponible para dispositivos con sistema operativo Windows, Linux, OS X, Android e iPhone OS. El protocolo que emplea para gestionar los documentos se denomina WebDAV (“Web Distributed Authoring and Versioning”), el cual explicaremos en los siguientes capítulos de esta memoria.

A parte de gestionar los archivos, Owncloud tiene la posibilidad de transmitir archivos de música, ver vídeos, visualizar archivos PDF (“portable document format”, formato de documento portátil) entre otras opciones. Además, integrará un calendario, agenda y un escaneo antivirus. El software incluye la administración de usuarios y grupos pudiéndoles otorgar o no derechos para editar y compartir archivos. A pesar de todo esto, Owncloud tiene algunos inconvenientes como la primera sincronización que es bastante lenta. También tiene algunos problemas de cifrado y si queremos editar o ver archivos de office necesitaremos integrar la aplicación Onlyoffice Developers. Para contratar este software existen dos licencias, una denominada “Standard” y otra “Enterprise”. Las diferencias más relevantes entre ellas son las limitaciones en el soporte, la posibilidad de migraciones desde otra aplicación, uso clientes móviles y la posibilidad de editar archivos de office. Si se precisara de asistencia profesional la suscripción estándar oscilaría entre 3k€ y 4,8k€ para 50 y 100 usuarios. En el caso de la Enterprise entre 7,2k€ y 11,5 k€ para 50 y 100 usuarios (precios aproximados).

Otra opción es el software Seafile. Es muy parecido al anterior, aunque escrito en el lenguaje de programación de sistemas (conocido como “C”) y Python. También está disponible para dispositivos Windows, Linux, OS X, Android, iPhone OS y Windows Phone. El protocolo que utiliza para acceder a los directorios de las carpetas es LDAP (“Lightweight Directory Access Protocol”). Tiene incluidas algunas características relevantes en temas de seguridad como el uso de las autenticaciones de dos factores, el control de acceso a carpetas o la encriptación de datos. Aunque no tiene tantas opciones como el anterior. Sus tipos de licencia son parecidos a Owncloud, aunque el precio es más elevado. Aproximadamente para una licencia de 9 usuarios con soporte de ayuda sólo por correo limitada serían 1k\$ al año aproximadamente.

Una alternativa al software Owncloud es NextCloud. Es muy parecido ya que lo crearon los mismos desarrolladores. También accede a la gestión de usuarios y carpetas utilizando el protocolo WebDAV (“Web Distributed Authoring and

Versioning”). Al igual que la primera opción de software está basada en Apache. Utiliza bases de datos en lenguaje PHP (“Hypertext Preprocessor”, lenguaje de programación para desarrollo web) y MySQL (“Structured Query Language”). Escrito en lenguajes de programación PHP y JavaScript. Además, puede soportar dispositivos con sistema operativo Windows, Linux, OS X, Android e iPhone OS. Incluye alguna característica adicional de seguridad a todas las que tiene Owncloud. Como la posibilidad de encriptar tanto en la transmisión como en el almacenamiento de los datos. También tiene integrada su propia API (“Application Programming Interface”). Para contratar este software existen tres tipos de licencia. La diferencia entre cada una de ellas es el soporte profesional que ofrecen.

En la Básica ofrecen soporte por correo electrónico un año, la Standard por teléfono durante tres años. Finalmente, la tarifa Premium tanto correo como llamadas por teléfono durante 10 años.

A continuación, mostraremos una tabla con precios aproximados:

Licencias/Usuarios	50 usuarios (€)	100 usuarios (€)
Basic	1,9k	3.4k
Standard	3.4k	6.1k
Premium	4.9k	8.9k

[Tabla 1: Licencias NextCloud]

La mejor opción de los softwares anteriores sería Owncloud o NextCloud. Ya que hoy en día las pequeñas las pequeñas y medianas empresas quieren reducir en costos. Ya que no tiene grandes presupuestos para invertir. Sus licencias tienen un coste inferior con un mayor número de usuarios a Seafile.

Entre esos dos softwares la mejor opción es NextCloud por diferentes motivos. El principal motivo es la mejora en la seguridad. Ha perfeccionado los procesos tanto de inicio como de transferencia de archivos. Como hemos indicado anteriormente, aplica la autenticación en dos pasos y la posibilidad de encriptación de archivos entre otros protocolos de seguridad que veremos en el capítulo de seguridad de esta memoria con más detalle. Además, siguen actualizando la documentación oficial y corrigiendo fallos o “bugs” que se encuentran los clientes en el software.

NextCloud sólo limita con sus licencias en la contratación del soporte. Sin embargo, Owncloud limita tanto en componentes como soporte en sus licencias. En el tipo de licencia Enterprise se podrán acceder a algunos servicios extras que en la Básica no existen. Como, por ejemplo, como la gestión de flujos de trabajo o la búsqueda en los textos. Además, todas las ediciones de NextCloud son públicas. En cambio, Owncloud tiene la edición Estándar con licencia pública pero la edición Enterprise está bajo una licencia comercial.

Otro de los motivos por el cual nos hace declinar la balanza por NextCloud es la consulta de archivos y algunas funcionalidades extras como las llamadas y videollamadas que Owncloud no dispone. NextCloud dispone gratuitamente del visor y editor de archivos de Office. Sin embargo, para Owncloud deberemos instalar una aplicación de pago.

En resumen, la mejor opción será el software NextCloud por la mejora de seguridad, sus pocas limitaciones, licencias de software libre, precio y compatibilidad con archivos Office. También cuenta con una interfaz gráfica, pero para nuestro proyecto no la vamos a utilizar por ello está deshabilitada. Para acceder a los archivos utilizamos su propia API (“Application Programming Interfaces”).

3.2 WordPress

La evolución de internet ha aumentado la participación de usuarios en blogs, redes sociales y páginas web. Gracias a esto, ha sido posible una evolución en la forma de crearlas. Desde hace unos años para poder crear un blog o una página web había que picar desde cero con varios lenguajes de programación. Unos más complejos que otros. Hoy en día, sin tener un gran conocimiento para implementar código existen aplicaciones de software que nos facilitan crear página web. Estas aplicaciones se denominan gestores de contenido o “Content Management System” (CMS). Los más prácticos y populares por los usuarios son: WordPress, TYPO3, Joomla, Drupal, Contao y Neos.

Todos son gestores CMS de código abierto, multiplataforma y bastante usados. La diferencia entre cada uno de ellos son los requisitos que se debe tener para instalarlos. Una de las características que los diferencian es el servidor y bases de datos compatibles. La tabla siguiente mostrará unas pequeñas diferencias entre algunos de ellos las cuales nos ayudarán a elegir el CMS:

Características:	WordPress	Typo3	Joomla	Neos
Dirigido a:	Blog web	Empresas	Web	Empresas
Servidor compatible	Web PHP MySQL	Apache NGINX MS IIS	Apache NGINX MS IIS	Apache NGiNX
Bases de datos compatibles	MySQL Maria DB	MySQL PostgreSQL Oracle Firebird MS SQL Server	MySQL MS SQL Server PostgreSQL	MySQL MariaDB PosgreSQL Oracle Microsoft
Lenguaje plantillas	PHP	TypoScript	PHP	Fluid AFX

[Tabla 2: Comparativa gestores de contenido]

Mayoritariamente los servidores web suelen soportar PHP y MySQL y las bases de datos son compatibles con MySQL. Por tanto, tendríamos varias opciones. Sin embargo, hemos utilizado WordPress. Aunque desde un primer momento fue desarrollado para blogs, cada vez más empresas lo utilizan para diseño de páginas web. La interfaz es muy intuitiva y cualquiera puede crear una web sin tener conocimiento. Incluso las empresas pueden crear una tienda online sin tener conocimientos avanzados para su uso. Gracias a la instalación de plugins, extensiones y su menú de administrador facilitan el diseño. Estos plugins se suelen actualizar continuamente, por lo que el software no se queda obsoleto. Se trabaja mediante una interfaz gráfica para ver los usuarios y un diseño del lado del servidor oculto donde lo pueden ver los administradores y editores del sitio. Todas estas ventajas no nos las proporciona ningún otro gestor de archivos.

Para usar WordPress tenemos dos opciones: WordPress.org o WordPress.com dependiendo del uso que le vayamos a dar. Si lo que queremos es crear una página web para nuestra empresa utilizaremos .org en el caso que queramos crear un blog gratuito utilizaremos .com. Obviamente para nuestro proyecto instalaremos.org. De esta forma podremos tener el control para publicar, dar de alta o baja usuarios, actualizar cualquier plugin o entrar mediante el gestor de archivos con una inversión baja.

La dirección web poder acceder al inicio de WordPress será <http://tu-sitio-wordpress.com/wp-admin> La interfaz gráfica consta de una zona de publicación (pública) y una zona de administración(privada).

En zona de administración se compone de una barra de herramientas con menús emergentes. Las opciones de esté menú son:

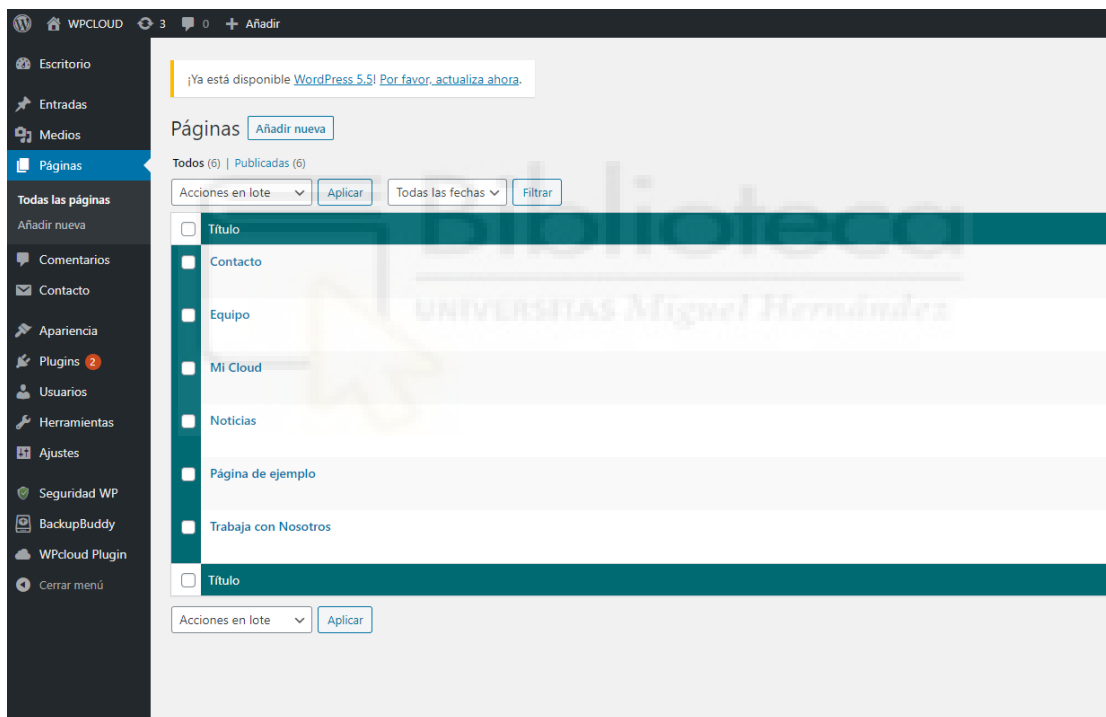
- **Entrada:** añadir/eliminar nuevas entradas, actualizar categorías o etiquetas.
- **Medios:** almacenamiento imágenes y archivos subidos.
- **Páginas:** añadir / eliminar páginas.
- **Apariencia:** cambiar/ crear menús, ubicación de menús, fondo página web.
- **Plugin:** instalar y gestionar plugins.
- **Usuarios:** crear/eliminar usuarios. WordPress tiene varios tipos de usuario: super administrador, administrador, colaborador, autor y editor.

El administrador, es el jefe, tiene acceso a toda la configuración de la página. Encargado de actualizaciones, instalación de plugins y configuración de software. También existe el rol de super administrador por si hubiera la necesidad de tener varios administradores. El suscriptor sólo puede acceder a contenido privado. Ideal para webs que permitan contenido sólo para usuarios registrados. El colaborador puede escribir

artículos y modificarlos, pero necesita la aprobación de un administrador para publicar. Si tiene los permisos para publicar se reconoce como usuario autor. Finalmente, el editor es como un administrador, pero no puede gestionar la web.

- **Herramientas:** configurar varias herramientas e importar el contenido de WordPress en un archivo.
- **Ajustes:** cambiar configuración cuenta de WordPress.

Dependiendo del tipo de usuario este menú podrá variar según los permisos. Para cualquier duda de usabilidad de WordPress podemos consultar una extensa guía actualizada desde la siguiente url: <https://wordpress.com/es/support/>.



[Figura 12: Menú administración WordPress]

Por otra parte, la zona de publicación se puede dividir en dos partes. Una barra de herramientas configurable y el espacio de las publicaciones.



[Figura 13: Barra herramientas y zona publicación de WordPress]

La barra de herramientas es totalmente configurable para poder acceder rápidamente a los menús más usados. Debajo tenemos la zona de las publicaciones. Esta zona está basada en dos conceptos: entradas y páginas. La diferencia entre ellas es que las páginas contienen información estática y las entradas son noticias o artículos, que suelen ir cambiando.

3.3 Instalación WordPress y NextCloud.

Una vez conocidas las dos herramientas vamos a instalarlas en el servidor. Suponiendo como se indicó en el primer capítulo de la memoria que tenemos el hosting contratado y el dominio. Podremos comenzar a descargar el software de WordPress desde su página web: <https://es.wordpress.org/>



[Figura 14: Página web WordPress]

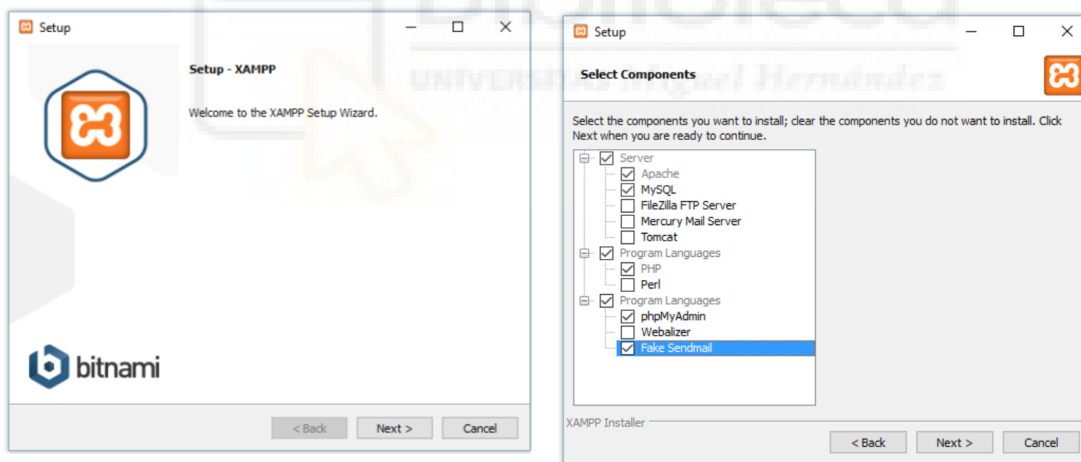
Pulsaremos sobre el botón “Consigue tu WordPress” y descargamos la última versión. Una vez realizado necesitamos un gestor de base de datos. WordPress trabaja con un sistema de base de datos denominado MySQL. Para trabajar con él, necesitamos instalar el paquete de software libre se denomina XAMPP. Cuyas siglas significan X cualquier sistema operativo, “A” de Apache, “M” de MariaDB/MySQL, “P” de PHP y “P” de Perl.

Entraremos en su página web y pulsaremos sobre el sistema operativo que necesitemos descargar: <https://www.apachefriends.org/es/index.html>



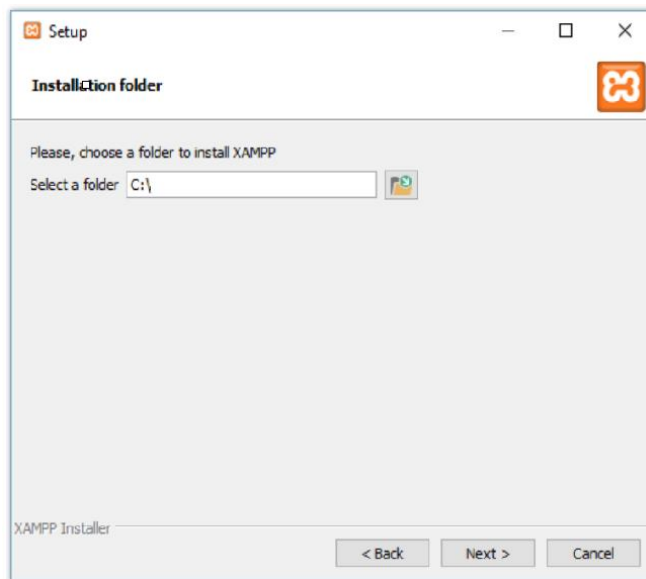
[Figura 15: Página web XAMPP]

Una vez descargado tendremos que ejecutarlo. Accederemos a las siguientes pantallas para configurarlo y deberemos pulsar en ambas en “Next”.



[Figura 16: Instalación XAMPP paso 1]

Seguidamente elegiremos la ubicación del archivo, que lo pondremos en el directorio raíz del servidor y pulsamos sobre “Next”.



[Figura 17: Instalación XAMPP paso 2]

Finalmente, para acabar pulsaremos Next y Finalizar en las siguientes pantallas.



Como anteriormente lo hemos guardado en el directorio raíz, ahora podremos entrar al navegador y nos aparecerá la siguiente pantalla:



Bienvenido a XAMPP para Linux 7.0.26

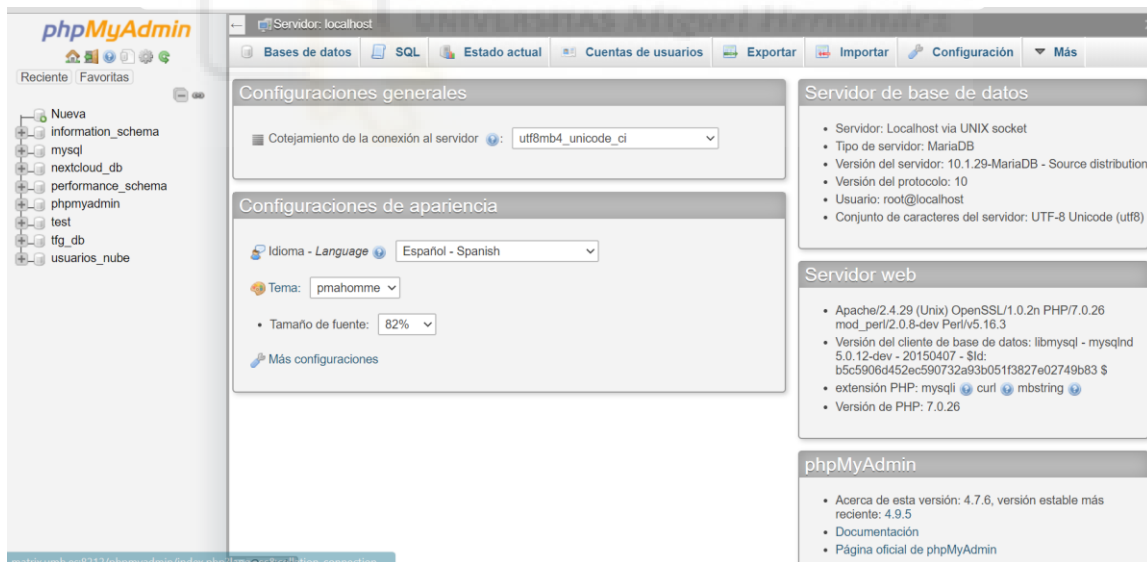
¡Has instalado con éxito XAMPP en este sistema! Ahora puede comenzar a usar Apache, MariaDB, PHP y otros componentes. Puede encontrar más información en la sección de Preguntas frecuentes o consultar las Guías de procedimientos para comenzar con las aplicaciones PHP.

XAMPP está destinado solo para fines de desarrollo. Tiene ciertas opciones de configuración que facilitan el desarrollo local pero que no son seguras si desea que otros puedan acceder a su instalación. Si desea tener su XAMPP accesible desde Internet, asegúrese de comprender las implicaciones y de haber consultado las preguntas frecuentes para aprender cómo proteger su sitio. Alternativamente, puede usar WAMP, MAMP o LAMP, que son paquetes similares que son más adecuados para la producción.

Inicie el Panel de control de XAMPP para verificar el estado del servidor.

[Figura 18: Instalación XAMPP paso 3]

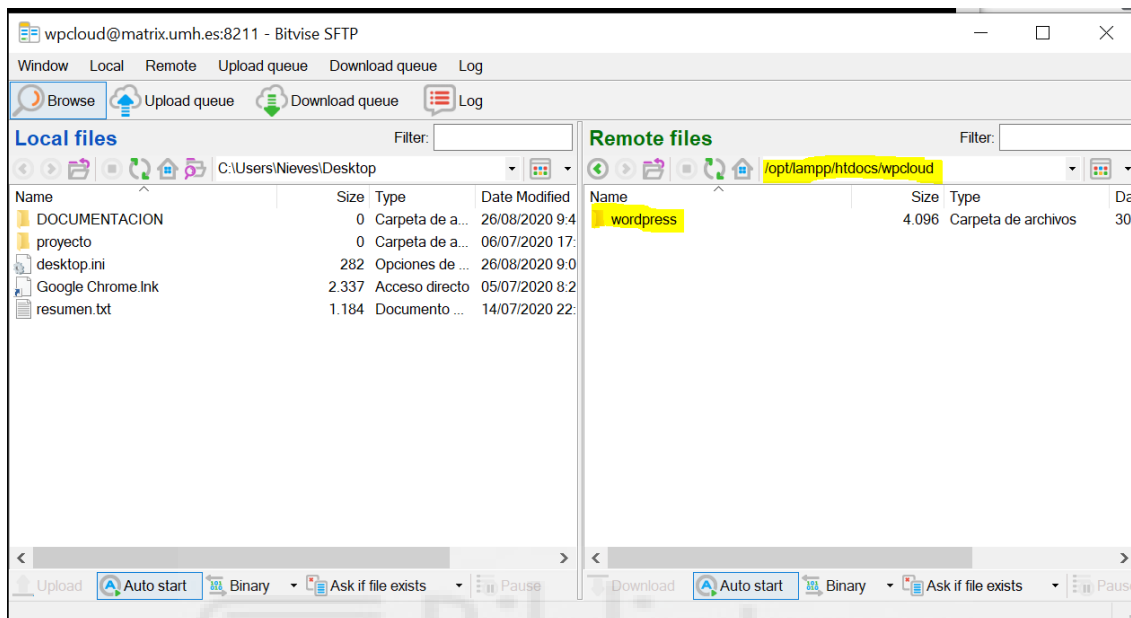
Podremos entrar a la configuración de las bases de datos pulsando sobre "phpAdmin".



[Figura 19: Instalación XAMPP paso 4]

El siguiente paso será instalar WordPress en Xampp. Para ello descomprimos el archivo de instalación de WordPress que nos habíamos descargado anteriormente.

La carpeta que se ha descomprimido la deberemos insertar en la siguiente ruta: “/opt/lampp/htdocs”. Aunque en nuestro caso hemos generado una carpeta denominada “wpcloud” y dentro lo hemos instalado, como se muestra en la siguiente imagen. Por si se quisiera instalar varias versiones de WordPress en un futuro.



[Figura 19: Instalación WordPress paso 1]

Abriremos el navegador y accederemos a la dirección web correspondiente. En nuestro caso <http://matrix.umh.es:8212/wpcloud/wordpress/>. Se nos abrirá una pantalla para comenzar la instalación de WordPress. Pulsaremos sobre “vamos a ello”.



[Figura 20: Instalación WordPress paso 2]

En la siguiente pantalla se mostrará un formulario. En el cual deberemos indicar el nombre de la base de datos, nombre y contraseña del usuario de WordPress,

el servidor, el prefijo de las tablas (normalmente “wp_” en próximos capítulos lo detallaremos). Lo rellenaremos y enviaremos. Y ejecutaremos la instalación.



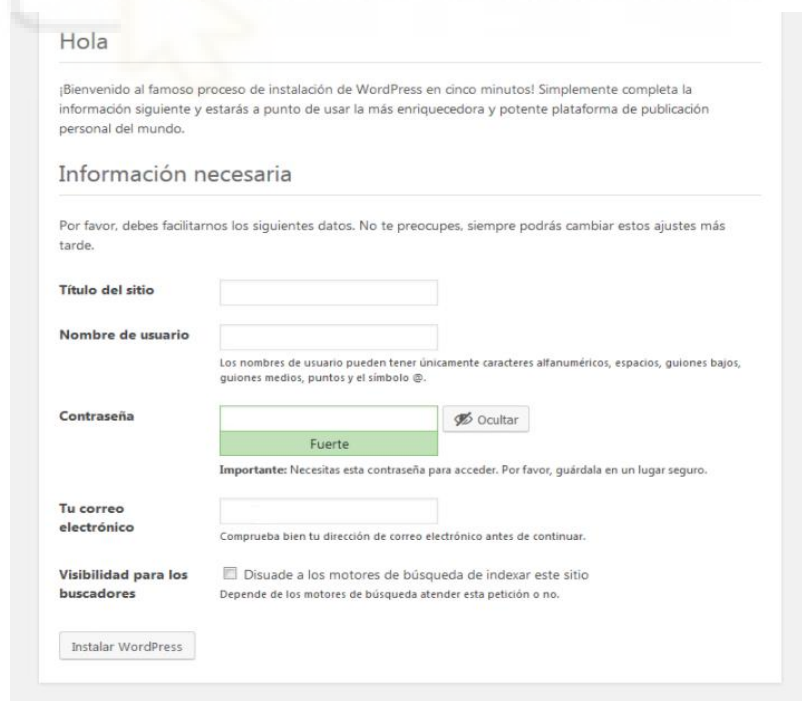
The screenshot shows the third step of the WordPress installation process. At the top center is the WordPress logo. Below it, a text block reads: "A continuación deberás introducir los detalles de conexión a tu base de datos. Si no estás seguro de esta información contacta con tu proveedor de alojamiento web." Below this are five input fields, each with a label and a descriptive note:

- Nombre de la base de datos:** El nombre de la base de datos que quieres usar con WordPress.
- Nombre de usuario:** El nombre de usuario de tu base de datos.
- Contraseña:** La contraseña de tu base de datos.
- Servidor de la base de datos:** Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona.
- Prefijo de tabla:** Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

At the bottom left, there is a button labeled "Enviar".

[Figura 21: Instalación WordPress paso 3]

Después de ejecutar la instalación deberemos indicar el título de la página, el nombre y contraseña de usuario puesto anteriormente y el correo electrónico. Pulsaremos sobre “Instalar WordPress”.




The screenshot shows the fourth step of the WordPress installation process. It starts with a greeting "Hola" and a welcome message: "¡Bienvenido al famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo." Below this is the section "Información necesaria". A note says: "Por favor, debes facilitarnos los siguientes datos. No te preocupes, siempre podrás cambiar estos ajustes más tarde." The form includes the following fields and options:

- Título del sitio:** A text input field.
- Nombre de usuario:** A text input field. A note below it says: "Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @."
- Contraseña:** A password input field with a strength indicator showing "Fuerte" in green and an "Ocultar" button.
- Tu correo electrónico:** A text input field. A note below it says: "Comprueba bien tu dirección de correo electrónico antes de continuar."
- Visibilidad para los buscadores:** A checkbox labeled "Disuade a los motores de búsqueda de indexar este sitio" with the note "Depende de los motores de búsqueda atender esta petición o no."

At the bottom left, there is a button labeled "Instalar WordPress".

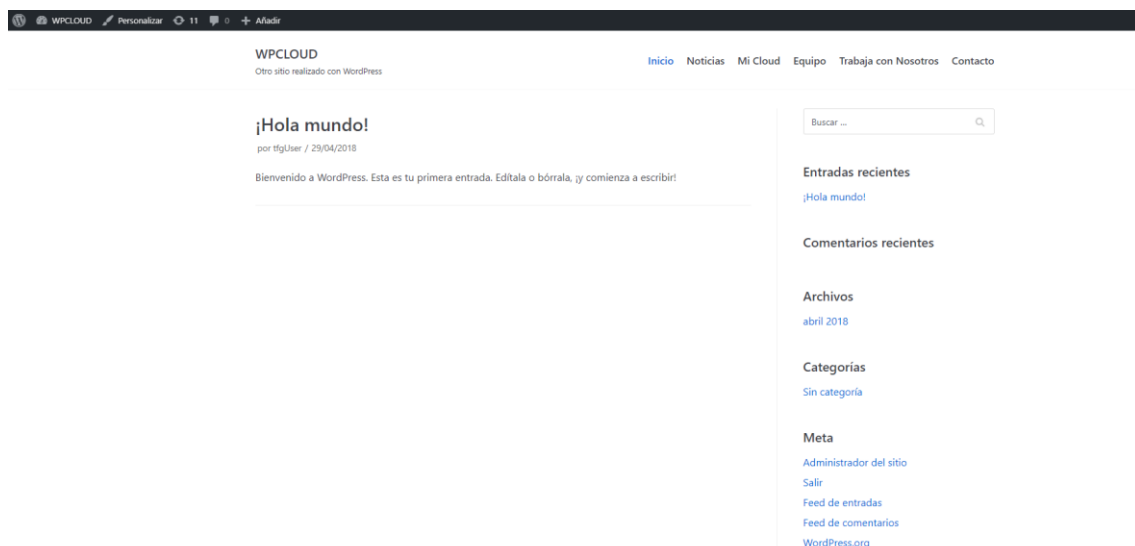
[Figura 22: Instalación WordPress paso 4]

Para comprobar nuestra instalación insertaremos nuestras credenciales en la ruta correspondiente de WordPress, para nuestro plugin la dirección web es: <http://matrix.umh.es:8212/wpcloud/wordpress/wp-login.php>



[Figura 23: Instalación WordPress paso 5]

Podremos comenzar a diseñar nuestra página web.



[Figura 24: Instalación WordPress paso 6]

Para la instalación de NextCloud deberemos acceder a la página web y descargarnos el software: <https://nextcloud.com/install/#instructions-server>

Archive File Web Installer Appliances X

The archive should be extracted in a folder your web server has access to. Latest stable version: 19.0.2 ([Changelog](#))

Download Nextcloud

Follow the [Nextcloud Admin Manuals](#) installation chapter.
If you already run Nextcloud, refer to the [upgrade manual](#).
[Need an enterprise solution?](#)

[Details and download options](#) ↓

Looking for older versions or major releases?

Nextcloud Server does not support Microsoft Windows. We recommend using a [virtual machine](#) or [docker image](#) on Windows Server.

Security note:
To receive information about updates and security issues, we recommend a subscription to our low-traffic [newsletter](#).

Release channels:
We offer [Release Channels](#) with production, stable, beta and daily-branches. This gives you the opportunity to choose your balance between stability and features.

[Figura 25: Instalación NextCloud paso 1]

Descomprimiremos e instalaremos la descarga en la ruta “/opt/lampp/htdocs”

wpcloud@matrix.umh.es:8211 - Bitvise SFTP

Window Local Remote Upload queue Download queue Log

Browse Upload queue Download queue Log

Local files Filter: **Remote files** Filter:

C:\Users\Nieves\Desktop /opt/lampp/htdocs

Name	Size	Type	Name	Size	Type
DOCUMENTACION	0	Carpeta de a...	admin	4.096	Carpeta de arc
peters-login-redirect	0	Carpeta de a...	dashboard	4.096	Carpeta de arc
proyecto	0	Carpeta de a...	img	4.096	Carpeta de arc
desktop.ini	282	Opciones de ...	nextcloud	4.096	Carpeta de arc
Google Chrome.lnk	2.337	Acceso directo	webalizer	4.096	Carpeta de arc
resumen.txt	1.184	Documento ...	wpcloud	4.096	Carpeta de arc
			applications.html	3.607	Microsoft Edge
			bitnami.css	177	Documento de
			descargarFichero.php	1.148	Archivo PHP
			favicon.ico	30.894	Icono
			index.php	260	Archivo PHP

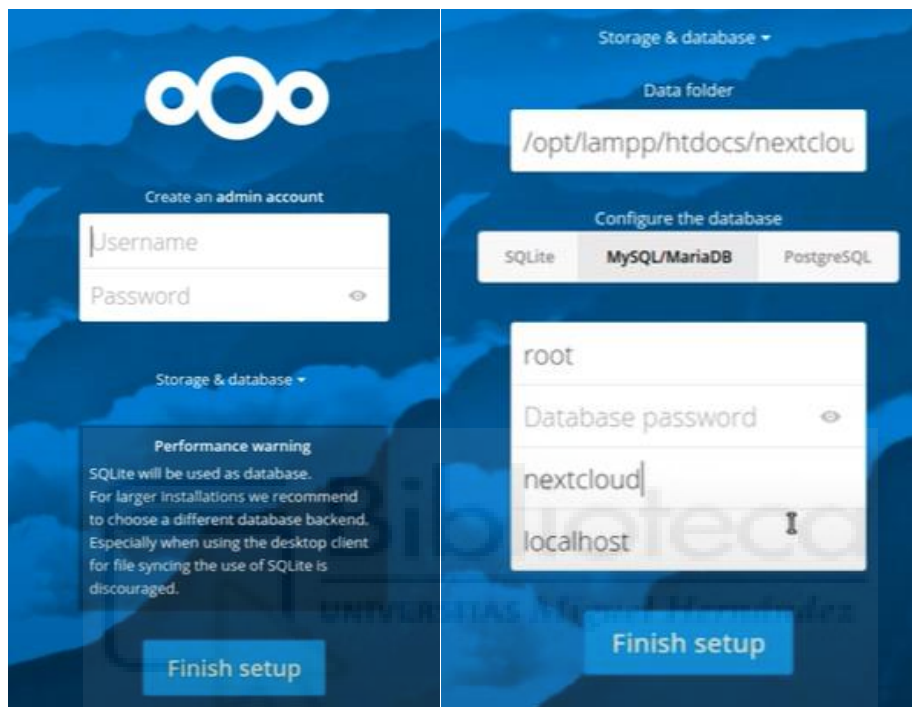
Upload Auto start Binary Ask if file exists Download Auto start Binary Ask if file exists

[Figura 26: Instalación NextCloud paso 2]

Para ejecutar la instalación entraremos en el navegador con esa ruta. En nuestro caso <http://matrix.umh.es:8212/nextcloud/index.php/login>. Como es la primera

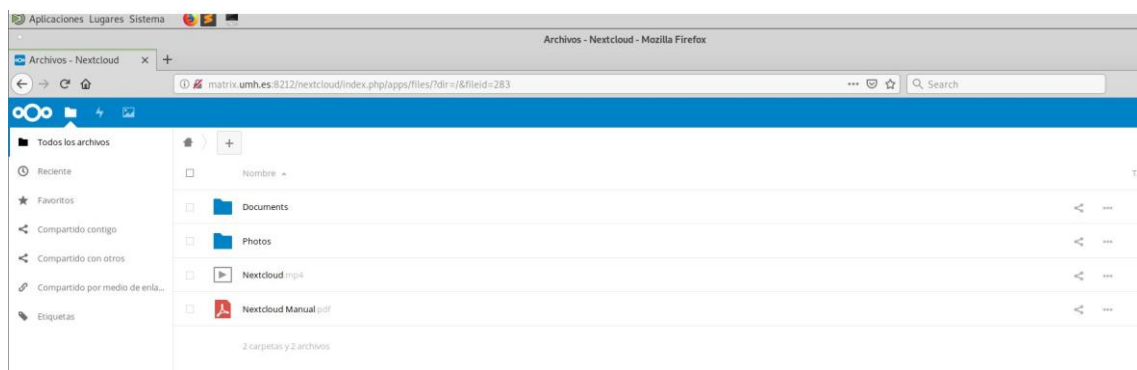
vez que vamos a acceder tendremos que configurar el usuario y la contraseña (imagen de la izquierda). Pulsaremos sobre “Finish setup”.

En el siguiente formulario configuramos la base de datos. En la imagen de la derecha tenemos un ejemplo. Data folder es una ruta que deberá salir por defecto. En nuestro caso es /opt/lampp/htdocs/nextcloud. Elegiremos la configuración como MySQL/MariaDB. Escribiremos el usuario y la contraseña de la base de datos. Y los dos siguientes campos los dejaremos por defecto como “nextcloud” y “localhost”.



[Figura 27: Instalación NextCloud paso 3]

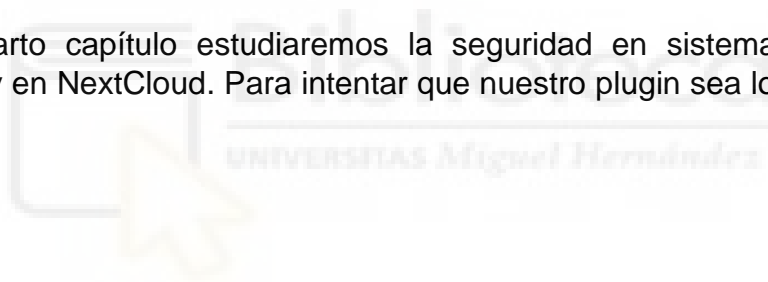
Esperaremos la finalización de la instalación y se iniciará la aplicación como vemos en la imagen.



[Figura 28: Instalación NextCloud paso 4]

Capítulo 4: Aspectos de Seguridad

En este cuarto capítulo estudiaremos la seguridad en sistemas cloud, en WordPress y en NextCloud. Para intentar que nuestro plugin sea lo más seguro posible.



4.1 Seguridad

El aumento del uso de sistemas clouds sobre todo en empresas a nivel mundial ha planteado a muchos ciberdelincuentes a planear ataques. Desde robo de contraseñas, pérdidas de información, envío de spam, suplantación de la identidad, APIs (“Application Programming Interfaces”) de conexión inseguras entre muchos otros.

La seguridad en un sistema cloud tiene como responsables tanto el proveedor de servicios como el cliente. El primero deberá asegurar que ninguna persona sin autorizar entre en el edificio donde tengan los servidores. Además, se deberá actualizar tanto el software como el hardware de sus equipos. También podrá segmentar los datos de un cliente. Es decir, los datos de un cliente los podrá tener fragmentados en varios servidores para añadir una mayor seguridad. El cliente por su parte deberá instalar parches de seguridad y mantener las políticas de seguridad tradicionales. Por ejemplo, borrar usuarios que no se utilicen, crear copias de seguridad, revisar el software, monitorizar todas las comunicaciones entre otras.

Para mejorar la seguridad podemos utilizar varios protocolos. El protocolo SSL (Secure Sockets Layer) y TLS (Transport Layer Security). Cuando se realiza una petición el servidor enviará el certificado. El navegador comprobará el certificado. En caso de que sea correcto enviará una clave cifrada. El servidor la deberá descifrar para establecer una conexión segura. De esta forma los datos transportados del servidor a la nube estarán cifrados.

También podemos utilizar el protocolo SSH (Secure Shell) y una conexión VPN (Virtual Private Network). El primero sirve para acceder a máquinas remotas a través de la red. El cliente pide acceder mediante el usuario, servidor y puerto. Si es correcto el servidor solicita una contraseña. Una vez el cliente la envía se puede establecer la conexión y se genera una clave de sesión entre el cliente y el servidor. La clave caduca cuando finalizan la sesión. Una VPN (Virtual Private Network) consiste en una conexión cifrada entre dos redes o un usuario y una red. Pudiendo mantener una comunicación con la nube segura.

Finalmente, si se necesita cualquier proveedor para gestionar el sistema deberemos firmar el Acuerdo a Nivel de Servicio (ANS). Es un documento donde se reflejan los requisitos de seguridad que deberán cumplirse tanto por el cliente como por el proveedor. Una vez conocidos algunos conceptos sobre seguridad, deberemos de conocer la seguridad en las dos herramientas que utilizamos a la hora de crear nuestro plugin. Tanto WordPress como en NextCloud.

4.2 Seguridad WordPress

Como hemos indicado en el apartado anterior, WordPress lo utiliza mucha gente alrededor de todo el mundo, por ello tienen tendencia a hacerle ciberataques. Es uno de los gestores de contenido más atacados. Obviamente nuestro servidor debe tener unas condiciones de seguridad pactadas con el proveedor. Aunque sin tener mucho conocimiento de programación podemos hacer que nuestra página web sea un poco menos accesible a los ciberdelincuentes. Seguidamente enunciamos una serie de medidas que deberemos implementar en nuestro plugin de WordPress:

- **Usuario Admin.**
El usuario administrador no debe contener ni el nombre de la empresa ni ser "admin". Es decir, no debe de ser un nombre deducible.
- **Contraseñas seguras.**
Los usuarios son obligados a introducir contraseñas con más de 8 caracteres incluyendo números, mayúsculas y símbolos especiales. Además, se debe implantar un sistema de política de contraseñas. Forzando a los usuarios a cambiar cada x tiempo su contraseña. También se pueden incluir sistemas de autenticación de varios factores o una contraseña de un solo uso ("OTP - One-Time Password").
- **Prefijo WP_ en tablas.**
Se cambia el prefijo de las tablas en base de datos "WP_" a cualquier otro. Aunque lo ideal sería generar prefijos aleatorios.
- **Permisos archivos y carpetas.**
Se deberá limitar los archivos "wp-config.php" y ".htaccess". En ellos están los accesos a la base de datos y a la configuración de WordPress. Para ello necesitaremos modificar los permisos de las carpetas con el código 755 y archivos con el código 644 a código 444 para que sólo sean de lectura.
- **Eliminar plugins y temas que no se usen.**
No deberemos tener temas o plugins instalados sino se usan. Se tienen instalados, pero no protegidos.
- **Limitar los accesos de login por fuerza bruta.**
Existen varios plugin como "*WP FAl2ban Redux*" o "*WP Fail2Ban*", los cuales no ayudarán a detectar ataques por fuerza bruta.
- **Actualizar WordPress, plugins y temas.**
Se deberían de revisar las actualizaciones manualmente del núcleo, plugins y temas.

- **Usar siempre SFTP.**

Las conexiones *FTP* (“File Transfer Protocol”) no son seguras se debe usar *SFTP* (“Secure File Transfer Protocol”) siempre que sea posible. Este método utiliza el protocolo *SSH* (“Secure Shell”) para añadir seguridad. Este protocolo utiliza una clave secreta en el mensaje, cliente y para el host como hemos estudiado en el apartado anterior.
- **Certificado SSL.**

El protocolo *SSL* (“Secure Socket Layer”) no es gratuito, aunque obligatorio. Encripta la información que se transmite entre el navegador y el servidor donde se aloje la web. Lo ideal sería contratar un certificado válido por una autoridad certificadora. Aunque existe la posibilidad de utilizar “Let’s Encrypt” que es una autoridad certificadora libre apoyada por la fundación de Linux. La entidad permite crear certificados web renovándose cada tres meses. Inclusive en WordPress existen plugins que nos facilitan la implementación del certificado como “WP Encrypt”, “WordPress Force HTTPS”.
- **Bloqueo de países.**

Bloquear con un plugin el acceso a otros países. Por ejemplo, países como Irán, China, Rusia son focos de ataques diarios constantes. Uno de los plugins que podríamos utilizar es “Admin Block Country”.
- **Eliminar el plugin Hello Dolly.**

Cuando se instala WordPress por defecto nos aparece un plugin denominado “Hello Dolly”. Este es muy usado por los hackers para insertar código malicioso, por tanto, deberemos eliminarlo.
- **Usar un Captcha.**

Cualquier identificación, registro, formulario que tengamos en la web debe de tener un Captcha. Un *Captcha* (“Completely Automated Public Turing Test to tell Computers and Humans Apart”) es un sistema para diferenciar entre un humano u ordenador. Esto nos ayudará a minimizar los ataques automáticos. Podemos instalar el plugin “Login No Captcha reCAPTCHA”
- **Backups.**

Las copias de seguridad son muy importantes. Aunque la mayoría de los proveedores de servicios se encarguen de estas copias deberíamos de tener las nuestras propias. Existen plugins para facilitarnos el trabajo. Uno de los plugins es “All in One WP Migration”. Además, a parte de ofrecer copias de seguridad también tiene funcionalidades de seguridad.
- **Versiones PHP.**

Deberíamos de ocultar la versión de “PHP”. Hay plugins en los navegadores como “Wappalyzer”. Pueden saber el tipo de servidor web que se está ejecutando y sus versiones. La mejor forma es programar WordPress para tener versión PHP como “ninguna”.

- **Rutas acceso.**
Renombrar las rutas de acceso mediante un plugin como “WPS Hide Login”.
- **Sesiones.**
Recordar cerrar las sesiones de los usuarios inactivos automáticamente. Además, recordar entrar con modo incógnito si el ordenador no es personal.
- **Registro Actividad.**
Mantener un registro de la actividad mediante un plugin por ejemplo “WP Security Audit Log”.
- **Bots.**
Añadir trampa para bots por ejemplo el plugin “BlackHole for Bad Bots”.
- **Descargas.**
Los plugins y temas tendremos que descargarlos de sitios fiables. Y comprobar en foros que no existe ningún bug.
- **Mover WP_config.php.**
Mover el archivo “wp_config.php” una carpeta superior. De esta manera el archivo estará en la raíz y nadie puede abrirlo mediante un enlace desde el navegador.
- **Edición de ficheros.**
No permitir la edición en los ficheros. Para ello deberemos incluir la siguiente línea de código en “wp_config.php”:

```
define ('DISALLOW_FILE_EDIT',true);
```

[Figura 29: Código seguridad WordPress paso 1]

- **Instalación plugins y temas.**
No permitir la instalación de plugins ni temas a no ser que sean instalados mediante la FTP. Para ello escribiremos la siguiente línea de código en “wp_config.php”:

```
define ('DISALLOW_FILE_MODS',true);
```

[Figura 30: Código seguridad WordPress paso 2]

- **Variable WP_DEBUG.**
Dentro de la configuración del archivo “wp_config.php” existe una variable global denominada “WP_DEBUG” para activar o desactivar el modo depuración. Mientras estemos en desarrollo tendrá el valor ‘true’.

Mediante la siguiente línea de código:

```
define ('WP_DEBUG',true);
```

[Figura 31: Código seguridad WordPress paso 3]

- **Deshabilitar la indexación y navegación de directorios.**
Para deshabilitar los directorios, deberemos poner al final del documento "htaccess" la siguiente línea:

```
Options -Indexes.
```

[Figura 32: Código seguridad WordPress paso 4]

Aunque posiblemente si tenemos un plugin de seguridad instalado ya nos lo haya ofrecido.

- **Deshabilitar trace o track.**
TRACE y TRAK son métodos del protocolo de transferencia de hipertexto abreviado. Se utilizan para depurar las conexiones del servidor web. Aunque pueden comprometer la seguridad del servidor. Pueden interceptar los datos manipulando estos dos métodos. Para prevenirlo deberemos deshabilitar estos dos métodos. Para ello en el archivo ".htaccess" deberemos incluir el siguiente código:

```
RewriteEngine on  
RewriteCond %{REQUEST_METHOD}^(TRACE|TRACK)  
RewriteRule .*- [F]
```

[Figura 33: Código seguridad WordPress paso 4]

- **Cabeceras.**

Hay varios tipos de cabecera que podemos añadir/quitar que harán más segura nuestra web.

Content-Security-Policy: se añaden para aumentar la seguridad en la web, aunque podría bloquear algún elemento importante como el registro de visitas de Google Analytics.

X-Content-Type: evita que se carguen hojas de estilos CSS (Cascading Style Sheets) o JavaScript maliciosas.

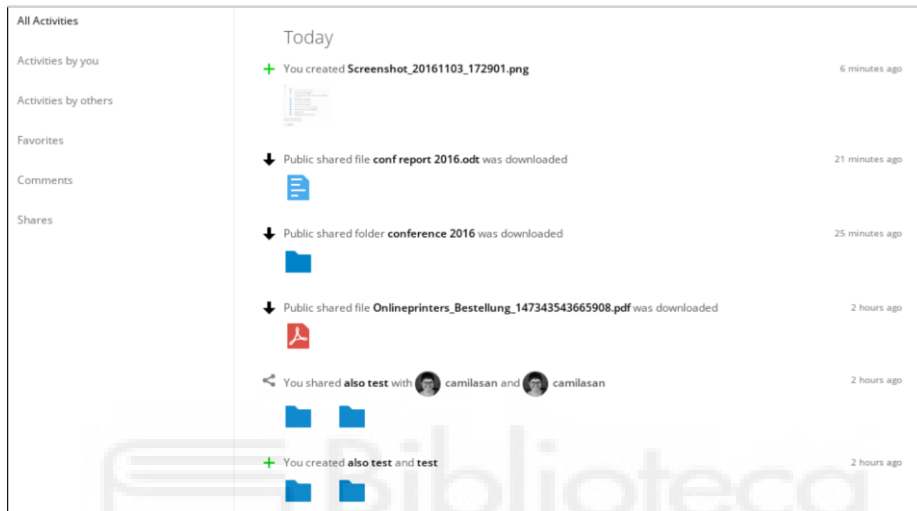
X-Frame: evita que nuestra página pueda ser abierta desde un dominio externo por medio de un *iframe*.

Cabecera XXS ("Cross-Site Scripting"): evita ataques de tipo XSS, bloqueando la carga de scripts externos.

4.3 Seguridad NextCloud

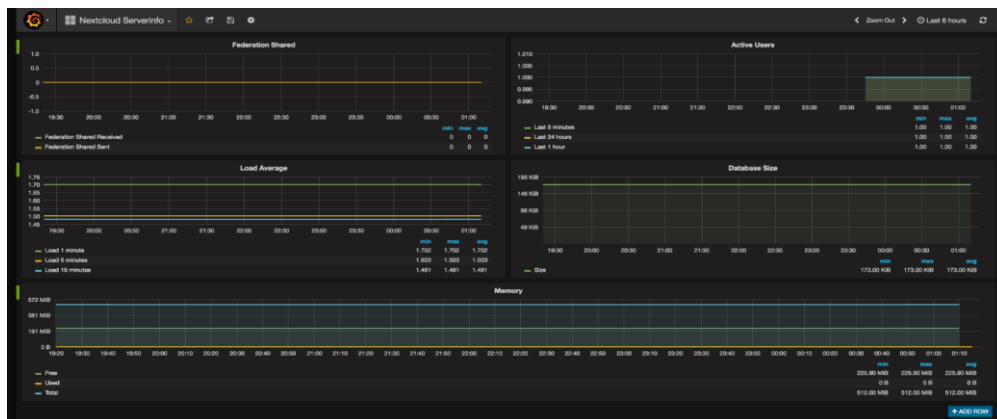
La seguridad es primordial para la compañía de NextCloud. Han incorporado herramientas extras de monitorización y rastreo. La aplicación contiene un historial de cambios recientes para llevar un control de los archivos modificados.

La plataforma se puede personalizar eligiendo los eventos que se quieren mostrar. Incluso programar para enviar un correo electrónico o un archivo feed RSS (“Really Simple Syndication”).



[Figura 34: Seguridad NextCloud – historial de cambios]

También los usuarios podrán supervisar información del servidor tanto el rendimiento como el estado del servidor de NextCloud. Incluyendo la posibilidad de consulta con un API (“Application Programming Interfaces”) para rastrear. Por ejemplo, podemos saber el número de usuarios que están activos, el tiempo de uso, sus acciones, ver estadísticas de almacenamiento entre otras cosas. Inclusive se pueden integrar algunas herramientas de gestión de redes como “OpenNMS” y herramientas para controlar grandes cantidades de datos generados en la máquina como “Splunk”.



[Figura 35: Seguridad NextCloud – integración aplicaciones de redes]

Puede ser útil ver información de los usuarios, sus conexiones, IP y el manejo de archivos entre otras acciones. Para rellenar informes de auditorías mediante el menú de Logging.

Level	Category	Message
Info	admin_audit	Login attempt: "jospoortvliet"
Info	admin_audit	Login attempt: "jospoortvliet"
Info	admin_audit	Login successful: "jospoortvliet"
Info	admin_audit	Login successful: "jospoortvliet"
Info	admin_audit	Login attempt: "jospoortvliet"
Info	admin_audit	Login successful: "jospoortvliet"
Info	admin_audit	Login successful: "jospoortvliet"
Info	admin_audit	Login attempt: "jospoortvliet"
Info	admin_audit	Login attempt: "jospoortvliet"
Info	cli	Memcache \OC\Memcache\APCu not available for distributed cache
Info	cli	Memcache \OC\Memcache\APCu not available for local cache
Info	cli	Memcache \OC\Memcache\APCu not available for distributed cache
Info	cli	Memcache \OC\Memcache\APCu not available for local cache
Info	admin_audit	File accessed: "/CamilaFiles/Notes/PersonalCare.ods"
Info	admin_audit	Preview accessed: "" (width: "32", height: "32" crop: "1", mode: "fill")
Info	admin_audit	Preview accessed: "/CamilaFiles/Notes/Tmp.md" (width: "32", height: "32" crop: "1", mode: "fill")
Info	admin_audit	Preview accessed: "/CamilaFiles/Notes/KnowledgeBase.md" (width: "32", height: "32" crop: "1", mode: "fill")
Info	admin_audit	Login successful: "jospoortvliet"
Info	admin_audit	Login successful: "jospoortvliet"

[Figura 36: Seguridad NextCloud – información para auditorías]

NextCloud permite compartir modificando los permisos para una mayor seguridad. Cualquier fichero se puede configurar para leer, crear o editar tanto individualmente como carpetas en grupos. Tener un control al compartir. Así cómo bloquear descargas y proteger con contraseñas. El gestor de archivos incluye una herramienta denominada "Flow". Esta utilidad permite automatizar tareas mediante un etiquetado a los archivos. Por ejemplo, convertir en PDF (Portable Document Format) o enviar un email cada vez que se cambie el fichero a un grupo u entre otras muchas restricciones.

Available flows

- Block access to a file
- Automated tagging
- PDF conversion
- More flows

Configured flows

When **File created, File updated, File renamed, Tag assigned** and Request URL matches Custom URL

→ **PDF conversion**
Convert documents into the PDF format on upload and write.
 Delete original, preserve existing PDFs

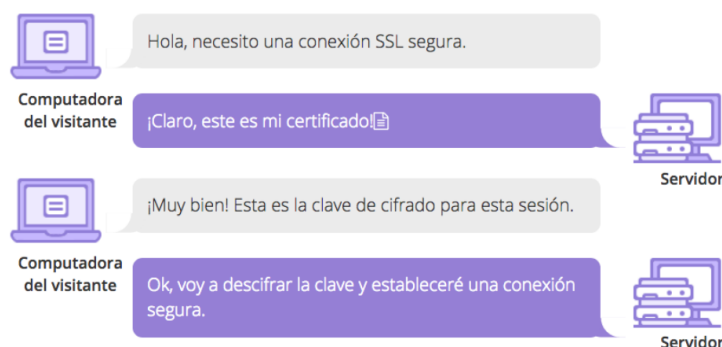
When **File is accessed** and File name is

→ **Block access to a file**

[Figura 37: Seguridad NextCloud – gestión archivos aplicación Flow]

Los archivos en NextCloud utilizan el cifrado con certificados SSL/TLS (“Secure Sockets Layer/Transport Layer Security” los mismos por desuso cambian el nombre). Estos se asocian a la identificación de la empresa mediante una clave criptográfica para para que no puedan ser descifrados.

Cuando alguien entra en una dirección web, se crea una clave de sesión válida por un tiempo limitado que se usa para codificar los datos que se transfieren.



[Figura 38: Seguridad NextCloud – protocolo “Secure Sockets Layer”]

Estos certificados conviven en las páginas web con el protocolo de transferencia de hipertexto abreviado (HTTP o HTTPS, hipertexto seguro). Utilizan este protocolo para enviar y recibir datos. También utiliza el cifrado “Rijndael”, es bastante complicado de atacar. Opcionalmente se puede incluir un cifrado extremo a extremo. En él sólo podrán leer el mensaje el emisor y receptor. Finalmente, este software libre facilita unas aplicaciones para cumplir con la ley GDPR o más conocida como la ley de protección de datos.

En resumen, NextCloud tiene todas las herramientas para poder hacer que esta herramienta sea segura, siempre que nuestro servidor lo sea.

Capítulo 5: Solución Integrada

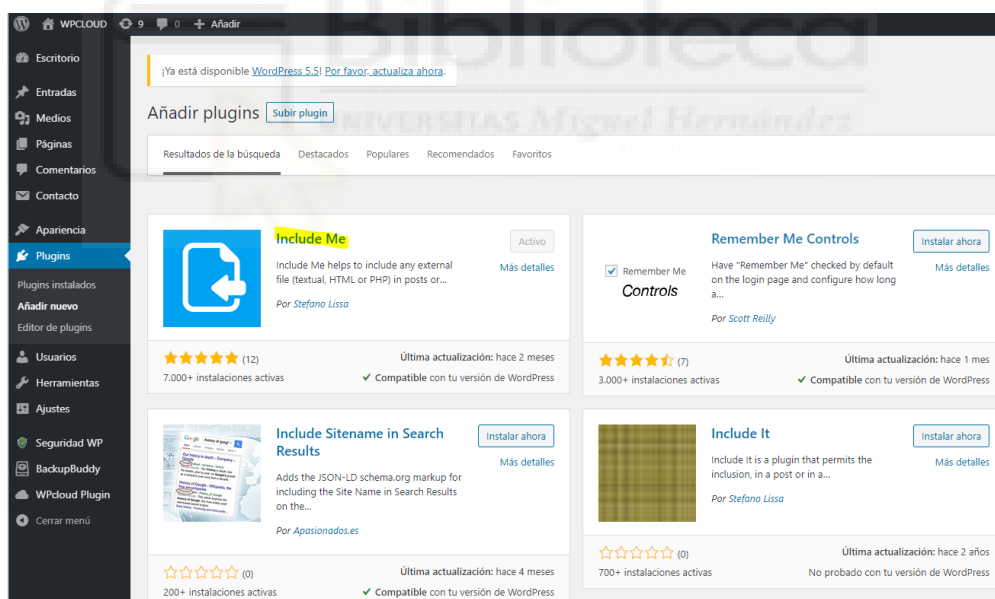
En este quinto capítulo estudiaremos dos tareas básicas. Necesarias para comenzar a desarrollar nuestro software. La primera tarea será implementar un plugin en WordPress. Para ello deberemos conocer el plugin “IncludeMe”, los ganchos o “hooks” y su API. La segunda tarea será estudiar el protocolo utilizado para poder comunicarnos con NextCloud y su API usando “WebDAV”.

5.1 WordPress Plugin

La solución para tener un software cloud para gestionar archivos en el propio servidor de los bufetes de abogados la desarrollaremos mediante un plugin en WordPress. Un plugin es un programa añadido a la web de WordPress. Para desarrollarlo deberemos conocer qué tipos de lenguaje deberemos utilizar.

El gestor de contenido está programado en PHP. (Hypertext Pre-Processor”). Es un lenguaje para crear páginas web. Por tanto, nuestro desarrollo deberá estar escrito en este lenguaje ya que necesitaremos utilizar su API. Sin embargo, vamos a necesitar un plugin denominado “IncludeMe”. Este software nos facilitará la ejecución de un archivo con extensión .php dentro de una página de WordPress.

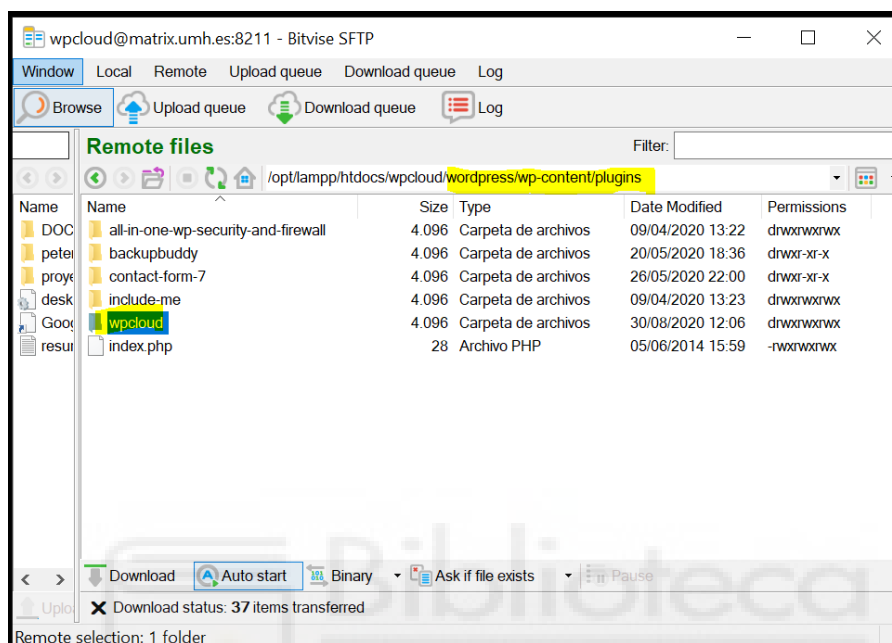
Tenemos dos formas diferentes para instalarlo. Entrando en la URL (Uniform Resource Location) siguiente <https://es.wordpress.org/plugins/include-me/> e instalándolo en la carpeta “plugins” dentro de WordPress mediante el programa “Bitwise”. Aunque otra opción es buscarlo e instalarlo mediante el menú de configuración de WordPress.



[Figura 39: Instalación Plugin IncludeMe]

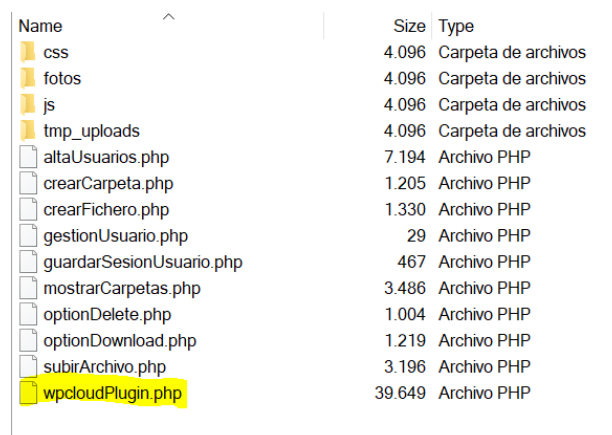
Una vez instalado crearemos una nueva página donde se visualizará nuestro código. Para utilizarlo deberemos usar el nombre del “shortcode” entre corchetes. En la página deberemos tener el siguiente código: “[micloud]”.

Seguidamente deberemos crear el archivo principal de nuestro software. Para ello, abriremos el programa “Bitwise SSH Client”. Accederemos a la ruta siguiente: ../wordpress/wp-content/plugins para crear el archivo. En nuestro caso lo denominamos “wpcloudPlugin.php”.



[Figura 40: Instalación plugin paso 1]

Crearemos una carpeta con el nombre del plugin denominada “wpcloud”. Dentro guardaremos un archivo “wpcloudPlugin.php”. En este archivo tendremos el código fuente de nuestra página web.



[Figura 41: Instalación plugin paso 2]

En el código declaramos la función de WordPress para poder añadir el plugin, “add_shortcode”. El primer parámetro con el nombre que referenciamos en la página y el segundo parámetro el nombre de la función.

Seguidamente crearemos la función del plugin. En nuestro caso se llamará “primerPlugin”.

```
add_shortcode( 'micloud', 'primerPlugin' );
```

```
function primerPlugin() {...}
```

[Figura 42: Instalación plugin paso 3]

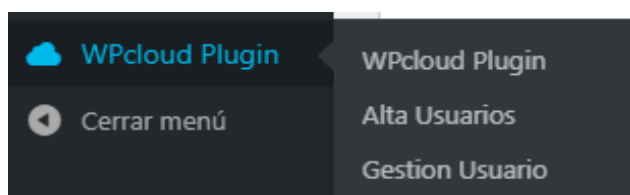
Adicionalmente en el caso de entrar en WordPress con un usuario administrador, nuestro plugin deberá tener un menú para gestionarlo. Para integrarlo aplicaremos cambios en la funcionalidad de nuestra página sin modificar el código fuente mediante “hooks o ganchos”. Para declarar el menú utilizaremos el gancho “add_action”. El primer parámetro es la etiqueta, es decir, el nombre de la acción a la que está enlazada. Y el segundo parámetro es el nombre de la función a llamar.

Se definirá la función “wp_cloud_create_admin_menu”. Añadirá los menús, submenús y sus correspondientes páginas necesarias para el panel de configuración. Utilizando la API de WordPress con las funciones “add_menu_page” y “add_submenu_page”.

```
function wpcloud_create_admin_menu() {  
    add_menu_page ( 'WPcloud Plugin', 'WPcloud Plugin',  
        'manage_options', 'wpcloud_create_admin_menu_plugin',  
        'wpcloud_create_admin_menu_function', 'dashicons-cloud' );  
  
    add_submenu_page ( 'wpcloud_create_admin_menu_plugin',  
        'Users', 'Alta Usuarios', 'manage_options',  
        'wpcloud_options_submenu1', 'wpcloud_options_function' );  
  
    add_submenu_page ( 'wpcloud_create_admin_menu_plugin', 'New User',  
        'Gestion Usuario', 'manage_options', 'wpcloud_options_submenu2',  
        'wpcloud_settings_function');  
}
```

[Figura 43: Instalación plugin paso 4]

Finalmente visualizamos el siguiente menú:



[Figura 44: Menú WPcloudPlugin]

En caso de no saber utilizar alguna de las funciones o ganchos podremos visitar la siguiente página web <https://developer.wordpress.org/reference/> donde está documentada toda la API (“Application Programming Interfaces”).

En nuestro desarrollo también usamos este tipo de ganchos para incluir códigos con extensión .js o .css. Suelen ser librerías para mejorar el diseño web.

```
//Añadir Bootstrap en la pagina principal
function bootstrap_css() {
    wp_enqueue_style( 'bootstrap_css',
        home_url().'/wp-content/plugins/wpcloud/css/bootstrap.min.css',
        array(), null);
}
add_action( 'wp_enqueue_scripts', 'bootstrap_css');

function bootstrap_js() {
    wp_enqueue_script( 'bootstrap_js',
        home_url(). '/wp-content/plugins/wpcloud/js/bootstrap.min.js',
        array('jquery'), null, true);
}
add_action( 'wp_enqueue_scripts', 'bootstrap_js');
```

[Figura 45: Código librerías]

Además del gancho “add_action” se utiliza “wp_enqueue_style”. Este “hook” se utiliza para poner en cola tanto estilos como scripts.

Para el proyecto hemos utilizado las siguientes librerías de las cuales haremos un resumen brevemente:

- **Bootstrap:** librería de software libre para el diseño de sitios web. Usado para cambiar la apariencia de contenedores, iconos, botones.
- **JQuery:** librería para gestionar eventos en JavaScript. Utilizada para programar las acciones de pulsar botones.
- **Cutefile:** librería para explorar archivos.

También hemos incluido los archivos CSS (Cascading Style Sheets) para definir los estilos.

5.2 NextCloud mediante API

El siguiente paso sería conocer las diferentes opciones que tiene la API (“Application Programming Interfaces”) para gestionar archivos. NextCloud utiliza el protocolo WebDAV (“Web based Distributed Authoring and Versioning”). Este protocolo facilita la transferencia de archivos por internet complementando al protocolo HTTP (“Hypertext Transfer Protocol”, muestra las páginas web). Por ejemplo, le permite crear y eliminar ficheros. Para utilizarlo en nuestro servidor necesitaremos instalar un “framework” (marco de trabajo) denominado “Sable/dav”. Podemos encontrarlo en el repositorio de github.com en la url: <https://github.com/sabre-io/dav>



[Figura 46: Logo Sable/dav]

Para su instalación bastará con descargarlo en la raíz de nuestro proyecto o mediante consola:

```
composer require sabre/dav ~3.2.0
```

[Figura 47: Instalación Sable/dav paso 1]

Añadiremos a nuestro código principal de la siguiente manera:

```
include '/home/wpcloud/vendor/autoload.php';
```

[Figura 48: Instalación Sable/dav paso 2]

NextCloud tiene una API (“Application Programming Interfaces”) para comunicarse con el cliente mediante este protocolo. Para implementarlo tendremos que utilizar una librería. Esta nos permite hacer peticiones HTTP (“Hypertext Transfer Protocol”) mediante el comando CURL (“Client URL”). Estas peticiones son mensajes enviados por el cliente para establecer una acción en el servidor. Algunos de ellos son listar archivos de una carpeta, descargar, subir archivos, crear carpetas, mover, copiar y eliminar archivos o carpetas. Siempre utilizaremos la dirección /nextcloud/remote.php/dav/ para realizar las peticiones.

Antes de implementar estas peticiones tendremos que conectarnos a WebDav de NextCloud. Para ello necesitaremos crear un objeto “client” de la siguiente manera:

```
$settings = array(  
    'baseUri' => 'localhost/nextcloud/remote.php/dav/',  
    'userName' => $userName,  
    'password' => $password  
);  
  
$client = new Sabre\DAV\Client($settings);
```

[Figura 49: Conexión cliente WebDav de NextcCloud]

Para tener un listado de fichero y carpetas de nuestro directorio de NextCloud utilizaremos el método PROPFIND. Con la ruta del directorio que queremos listar las carpetas y un array con las propiedades que queremos que nos devuelva como en el ejemplo:

```
$folder_content = $client->propFind($urlDirectory, array(
    '{DAV:}getlastmodified',
    '{DAV:}getcontenttype',
    '{http://owncloud.org/ns}favorite',
    '{http://owncloud.org/ns}has-preview'
),1);
```

[Figura 50: Listado clientes NextCloud]

Otras propiedades que podemos consultar son:

- {DAV:}getlastmodified
- {DAV:}getetag
- {DAV:}getcontenttype
- {DAV:}resourcetype
- {DAV:}getcontentlength
- {http://owncloud.org/ns}id The fileid namespaced by the instance id, globally unique
- {http://owncloud.org/ns}fileid The unique id for the file within the instance
- {http://owncloud.org/ns}favorite
- {http://owncloud.org/ns}comments-href
- {http://owncloud.org/ns}comments-count
- {http://owncloud.org/ns}comments-unread
- {http://owncloud.org/ns}owner-id The user id of the owner of a shared file
- {http://owncloud.org/ns}owner-display-name The display name of the owner of a shared file
- {http://owncloud.org/ns}share-types
- {http://owncloud.org/ns}checksums
- {http://owncloud.org/ns}has-preview
- {http://owncloud.org/ns}size Unlike `getcontentlength`, this property also works for folders reporting the size of everything in the folder.

[Figura 51: Propiedades usuarios]

En el siguiente apartado explicaremos con más detalle las peticiones. Aunque podemos adelantar que la usaremos para crear carpetas, visualizar carpetas, crear fichero y borrar fichero.

NextCloud también contiene una API (“Application Programming Interfaces”) de OCS (“Open Collaboration Service”). Todas las solicitudes que se quieran hacer deberán de utilizar el método CURL (“Client URL”) especificando el método de solicitud como por ejemplo “get”o “put”. Las solicitudes deberán contener un encabezado, autenticación básica o un conjunto de cookies de sesión válidas. Además del establecimiento de un cuerpo.

En nuestro desarrollo la utilizaremos para crear usuarios, compartir carpetas, subir y descargar archivos, cómo veremos en el capítulo siguiente.

Capítulo 6: Plugin WPCloud

En el sexto capítulo conoceremos la funcionalidad del software WPCloud que se ha desarrollado.



6.1 PLUGIN WPCLOUD

Mediante el plugin, los bufetes de abogados podrán compartir con sus clientes y viceversa ficheros. El plugin lo podemos fraccionar en cinco partes: acceso “Mi Cloud”, alta de usuarios, visualización de carpetas, herramientas y gestión de permisos.

6.2 ACCESO “MI CLOUD”

Para acceder al software cloud tanto clientes como abogados del bufete deberán entrar en el menú Mi Cloud. Este se creará cuando instalemos el plugin.

[Inicio](#) [Noticias](#) [Mi Cloud](#) [Equipo](#) [Trabaja con Nosotros](#) [Contacto](#)

[Figura 52: Menú Mi Cloud paso 1]

Accederemos a una página donde tenemos un botón de acceso al cloud. Pulsaremos sobre “Entrar”.

Mi Cloud

ENTRAR

[Figura 53: Página Mi Cloud paso 2]

Se nos abrirá una pantalla modal para insertar las credenciales. Insertamos el usuario y contraseña y pulsaremos sobre “Entrar”.

WPCLOUD
Otro sitio realizado con WordPress

Mi Cloud

ENTRAR

MICLOUD

Usuario: usuariq

Contraseña: ●●●●●●

ENTRAR CERRAR

[Figura 54: Acceso MiCloud paso 3]

Tanto el usuario como la contraseña será la misma que en WordPress. Sólo podrán acceder a “MICLOUD” aquellos clientes que estén dados de alta en el servicio. El abogado tendrá que dar de alta a sus clientes para que puedan acceder. En el caso que un cliente ya tenga usuario en WordPress, pero quiere tener los servicios del cloud. El abogado deberá darlo de alta igualmente pero su contraseña cambiará obligatoriamente. Por seguridad la sesión caerá si no existe actividad en menos de una hora y deberemos volver a conectarnos.

Para implementar un formulario de “login” alternativo al que tiene WordPress, hemos creado nuestro propio formulario redirigiéndolo a la URL del “login” de WordPress como se muestra en la imagen siguiente. De esta manera cuando pulsemos a “Entrar” se nos conectará directamente con la página del usuario en Wordpress.

```
<div class="modal dialog">  
<form id="formloginNew" action="'.home_url().'/wp-login.php" method="post">
```

[Figura 55: Implementación acceso MiCloud paso 4]

También hemos deshabilitado el nuestro menú de administración del plugin si el usuario no es un administrador. Para conocer el rol del usuario hemos utilizado la API de WordPress como se muestra en la imagen.

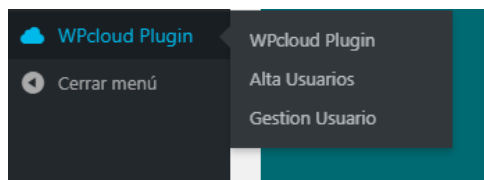
```
global $current_user;  
get_currentuserinfo();  
$usuario = esc_attr($current_user->user_login);  
$type = esc_attr($current_user->roles[0]);
```

[Figura 56: Implementación acceso MiCloud paso 5]

WordPress posee una variable `$current_user` que nos proporciona información sobre los usuarios y una función `get_currentuserinfo()` el cual recupera el objeto usuario conectado. La propiedad rol del usuario la recogemos mediante la función `esc_attr()`.

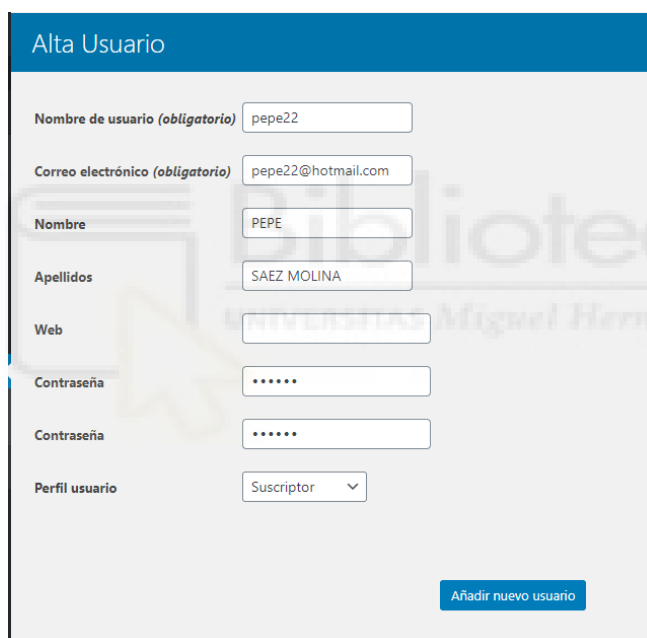
6.3 ALTA USUARIOS

Para dar de alta a los usuarios deberá insertar sus credenciales, ir al menú de Administración y seleccionar el submenú “Alta Usuarios”.



[Figura 57: Menú Alta Usuarios paso 1]

Aparecerá un formulario que se deberá rellenar.

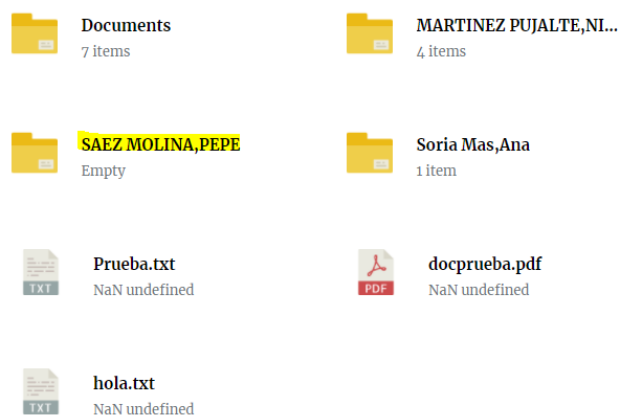
A screenshot of a web form titled 'Alta Usuario'. The form has a blue header bar with the title. Below the header, there are several input fields: 'Nombre de usuario (obligatorio)' with the value 'pepe22', 'Correo electrónico (obligatorio)' with 'pepe22@hotmail.com', 'Nombre' with 'PEPE', 'Apellidos' with 'SAEZ MOLINA', 'Web' (empty), 'Contraseña' (two fields with masked characters '*****'), and 'Perfil usuario' with a dropdown menu showing 'Suscriptor'. A blue button labeled 'Añadir nuevo usuario' is located at the bottom right of the form. A large, semi-transparent watermark 'Biblioteca Universitaria Miguel Hernández' is visible in the background.

[Figura 58: Formulario Alta Usuarios paso 2]

Pulsaremos sobre “Añadir nuevo usuario”.

Una vez realizado, se creará en el escritorio de cloud una carpeta con el nombre del cliente. La nomenclatura que tendrán las carpetas son Apellidos, Nombre.

files



[Figura 59: Carpeta usuario paso 3]

Ya tendríamos creado un nuevo usuario. Internamente, el código al pulsar “añadir usuario” realizará varias funciones. Por un lado, se creará el usuario en WordPress, en NextCloud, creará la carpeta y compartirá la carpeta para que el cliente también lo pueda ver.

Crear un usuario en WordPress.

Utilizaremos la función `wp_insert_user()` propia del gestor de contenido siendo la variable `$datos` un array de lo recogido por el formulario.

```
$user_id = wp_insert_user($datos);
```

[Figura 60: Creación usuario WordPress]

Crear un usuario en NextCloud.

Utilizamos el intérprete de comandos Curl. Como se muestra en la siguiente imagen, inicializaremos la sesión de curl mediante la función `curl_init()`. Seguidamente configuraremos la transferencia utilizando `curl_setopt_array()`, a esta función le pasaremos el objeto que devuelve la inicialización y un array con los parámetros de configuración. El array de configuración tiene incluida la dirección de la dirección web a capturar, el usuario y contraseña de NextCloud, la cabecera http, método de envío, y los datos a enviar que en este caso son el usuario y la contraseña. La dirección que utilizaremos para hacer la petición es `"/nextcloud/ocs/v1.php/cloud/users"`. Para comprobar que la conexión ha sido correcta utilizaremos `curl_exec()`. En el caso que sea true la conexión ha sido correcta y compartiremos la carpeta con el usuario.

Finalmente, utilizaremos `curl_close()` para cerrar la sesión curl.

```
$params=['userid'=> $userwpccloud, 'password'=>$passwpccloud . $srccloud];
$defaults = array(
    CURLOPT_URL => 'http://matrix.umh.es:8212/nextcloud/ocs/v1.php/cloud/users',
    CURLOPT_USERPWD => $_COOKIE["cookieUser"]. ':' . $_COOKIE["cookiePassword"] . $srccloud,
    CURLOPT_HTTPHEADER => array('OCS-APIRequest:true'),
    CURLOPT_POST => true,
    CURLOPT_POSTFIELDS => $params,
);
$ch = curl_init();
curl_setopt_array($ch, $defaults);

if(curl_exec($ch) === false)
{
    $resultcreateuser==array( 'message' => 'NextCloudError');
}
else
{
    $resultcreateuser=array( 'message' => 'Creado');
    CompartirCarpetaUsuario($userfolder,trim($userwpccloud));
}

curl_close($ch);
```

[Figura 61: Crear usuario NextCloud]

Crear carpeta cliente.

Para crear una carpeta utilizaremos el API de NextCloud de WebDAV. Estableceremos una conexión y utilizaremos el método MKCOL.

```
$settings = array(
    'baseUri' => 'localhost/nextcloud/remote.php/dav/',
    'userName' => $userName,
    'password' => $password
);

$client = new Sabre\DAV\Client($settings);
$response = $client->request("MKCOL", $urlD);
```

[Figura 62: Crear carpeta cliente]

Compartir carpeta:

A parte de las funciones para conectar y desconectar la sesión de curl, utilizaremos `curl_setopt_array()` para configurar la transferencia. La dirección de la petición será a “nextcloud/ocs/v2.php/apps/files_sharing/api/v1/shares”.

Y los parámetros que enviaremos serán el nombre de la carpeta o archivo, el tipo de archivo (siendo 0 usuario, 1 grupo, 3 enlace público, 6 recurso), usuario, permisos (siendo 1 leer, 2 actualizar, 4 crear, 8 eliminar, 16 compartir, 31 todos).

```
$namefolder="/" . trim($namesharedFolder);
echo 'Carpeta a compartir' . $namefolder . '<br/>';
echo 'Usuario' . $userid . '<br/>';

$params=['path' => $namefolder,'shareType'=> 0,'shareWith'=> $userid, 'permissions'=>31];
$defaults = array(
    CURLOPT_URL => 'http://matrix.umh.es:8212/nextcloud/ocs/v2.php/apps/files_sharing/api/v1/shares',
    CURLOPT_USERPWD => $_COOKIE["cookieUser"]. ':' . $_COOKIE["cookiePassword"] . $srccloud,
    CURLOPT_HTTPHEADER => array('OCS-APIRequest:true'),
    CURLOPT_POST => true,
    CURLOPT_POSTFIELDS => $params,
);

$ch = curl_init();
curl_setopt_array($ch, $defaults);

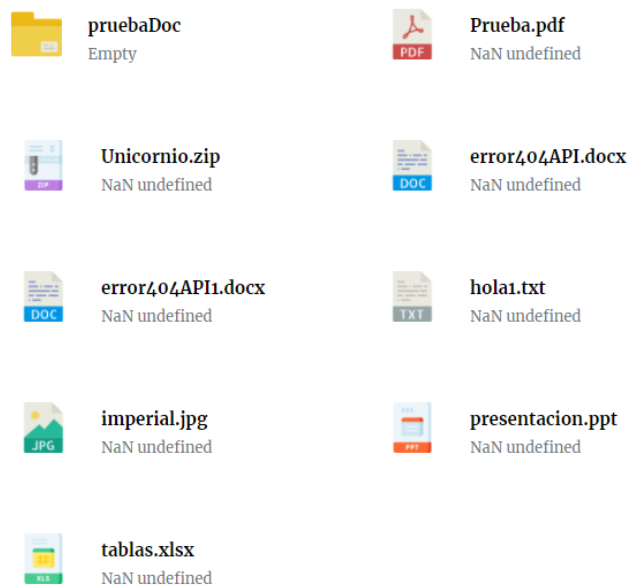
if(curl_exec($ch) === false){
    $resultcreateuser=array( 'message' => 'NextCloudError');
}else{
    $resultcreateuser=array( 'message' => 'Creado');
}
curl_close($ch);
```

[Figura 63: Compartir carpeta]

6.4 VISUALIZACIÓN DE CARPETAS

En la zona de publicación de WordPress nos deberán aparecer las carpetas que cada usuario tiene en su directorio. El software reconocerá aquellos ficheros que tengan extensión .zip, .doc, .jpg, .xlsx, .pdf, .docx, .txt, .ppt, .mp3 y .mp4. Como se puede ver en la imagen.

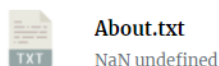
files → Documents



[Figura 64: Visualización carpetas paso 1]

Podemos entrar en cada una de las carpetas haciendo clic sobre ellas. Y para volver a los directorios podemos usar las etiquetas de nombres de los directorios. Por ejemplo, si estuviéramos dentro de esa carpeta con la ruta: “files -> MARTINEZ PUJALTE, NIEVES MARINA -> Documents” como se muestra en la imagen.

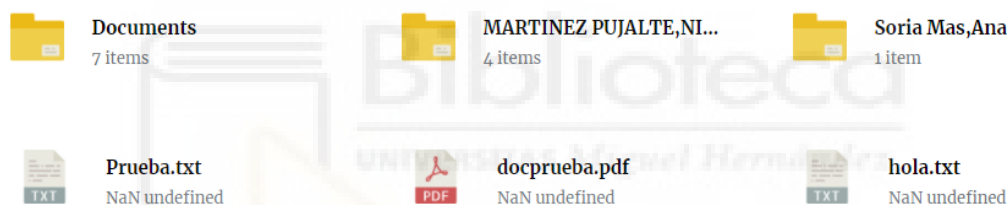
files → MARTINEZ PUJALTE, NIEVES MARINA → Documents →



[Figura 65: Visualización carpetas paso 2]

Y quisiéramos volver al directorio principal “files” podemos hacer clic en “files” y directamente nos aparecerá la pantalla:

files



[Figura 66: Visualización carpetas paso 3]

Para ayudarnos a visualizar las carpetas hemos instalado una librería denominada Cute File Browser, que simularía un explorador de archivos. La cual podremos descargar en la siguiente dirección web: <https://tutorialzine.com/2014/09/cute-file-browser-jquery-ajax-php>



[Figura 67: Cute File Browser paso 4]

Para poder dibujar las carpetas necesita un formato JSON (“JavaScript Object Notation”) específico. Un formato de ejemplo, lo podemos observar en la siguiente imagen.

```
{
  "name": "Documents", "type": "folder", "path": "files/Documents", "items": [
    {
      "name": "Prueba.pdf", "type": "file", "path": "files/Documents/Prueba.pdf", "items": []
    },
    {
      "name": "Unicornio.zip", "type": "file", "path": "files/Documents/Unicornio.zip", "items": []
    },
    {
      "name": "error404API.docx", "type": "file", "path": "files/Documents/error404API.docx", "items": []
    },
    {
      "name": "error404API1.docx", "type": "file", "path": "files/Documents/error404API1.docx", "items": []
    },
    {
      "name": "hola1.txt", "type": "file", "path": "files/Documents/hola1.txt", "items": []
    },
    {
      "name": "imperial.jpg", "type": "file", "path": "files/Documents/imperial.jpg", "items": []
    },
    {
      "name": "presentacion.ppt", "type": "file", "path": "files/Documents/presentacion.ppt", "items": []
    },
    {
      "name": "pruebaDoc", "type": "folder", "path": "files/Documents/pruebaDoc", "items": [
        {
          "name": "tablas.xlsx", "type": "file", "path": "files/Documents/pruebaDoc/tablas.xlsx", "items": []
        }
      ]
    },
    {
      "name": "MARTINEZ%20PUJALTE%2cNIEVES%20MARINA", "type": "folder", "path": "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA", "items": [
        {
          "name": "Carpeta Vacía", "type": "folder", "path": "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA/Carpeta Vacía", "items": [
            {
              "name": "Prueba%20Subida.pdf", "type": "file", "path": "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA/Carpeta Vacía/Prueba%20Subida.pdf", "items": []
            },
            {
              "name": "Unicornio.zip", "type": "file", "path": "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA/Carpeta Vacía/Unicornio.zip", "items": []
            }
          ]
        },
        {
          "name": "Documents", "type": "folder", "path": "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA/Documents", "items": [
            {
              "name": "About.txt", "type": "file", "path": "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA/Documents/About.txt", "items": []
            },
            {
              "name": "Nextcloud%20Manual.pdf", "type": "file", "path": "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA/Nextcloud%20Manual.pdf", "items": []
            },
            {
              "name": "PruebaBorrar.txt", "type": "file", "path": "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA/PruebaBorrar.txt", "items": []
            }
          ]
        }
      ]
    }
  ]
}
```

[Figura 68: Array específico paso 5]

Para conseguir formar ese formato JSON (“JavaScript Object Notation”) necesitamos conseguir recoger todos los elementos de los directorios. Para ello, conectaremos con el cliente DAV de NextCloud.

```
$settings = array(
    'baseUri' => 'localhost/nextcloud/remote.php/dav/',
    'userName' => $userName,
    'password' => $password
);

$client = new Sabre\DAV\Client($settings);
return $client;
```

[Figura 69: Visualización carpetas paso 6]

Crearemos una función denominada *propFindDirectory*. La cual mediante la función Propfind de Sable/dav listará todos los elementos de un directorio.

```
function propFindDirectory($client, $urlDirectory) {
    $folder_content = $client->propFind($urlDirectory, array(
        '{DAV:}getlastmodified',
        '{DAV:}getcontenttype',
        '{http://owncloud.org/ns}favorite',
        '{http://owncloud.org/ns}has-preview'
    ), 1);
    return json_encode($folder_content);
}
```

[Figura 70: Visualización carpetas paso 7]

Para conseguir lo que necesitamos deberemos realizar una función recursiva.

```
{name: "files", type: "folder", path: "files", items: Array(7)}
  items: Array(7)
    0: {name: "Documents", type: "folder", path: "files/Documents/", items: Array(9)}
    1: {name: "MARTINEZ%20PUJALTE%2cNIEVES%20MARINA", type: "folder", path: "files/MARTINEZ%20PUJALTE%2cNIEVES%20MARINA/", items: Array(4)}
    2: {name: "Prueba.txt", type: "file", path: "files/Prueba.txt", items: Array(0)}
    3: {name: "SAEZ%20MOLINA%2cPEPE", type: "folder", path: "files/SAEZ%20MOLINA%2cPEPE/", items: Array(0)}
    4: {name: "Soria%20Mas%2cAna", type: "folder", path: "files/Soria%20Mas%2cAna/", items: Array(1)}
    5: {name: "docprueba.pdf", type: "file", path: "files/docprueba.pdf", items: Array(0)}
    6: {name: "hola.txt", type: "file", path: "files/hola.txt", items: Array(0)}
    length: 7
    __proto__: Array(0)
  name: "files"
  path: "files"
  type: "folder"
  __proto__: Object
```

[Figura 71: Array directorio paso 8]

En esta función recursivas comprobaremos si el último carácter del path es /, si es afirmativo es porque nos encontramos ante una carpeta con contenido dentro. Entraremos un nivel, es decir, dentro de la carpeta y volveremos a llamar a la función *propFindDirectory*. Volveremos a realizar el mismo proceso hasta que completemos todos los directorios.

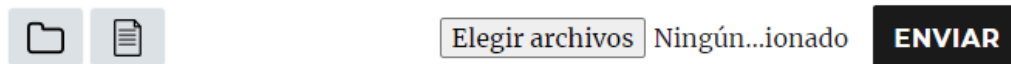


6.5 HERRAMIENTAS

Las herramientas son las acciones que va a poder hacer un usuario. Las acciones que tenemos programadas son:

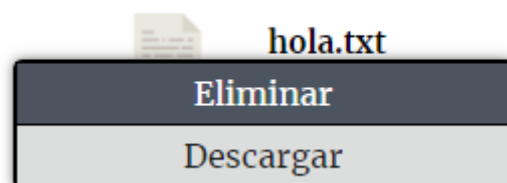
- Crear carpeta
- Crear fichero
- Subir fichero
- Borrar fichero
- Descargar archivos

Estas acciones están repartidas entre la barra de herramientas y el botón derecho. Por un lado, tenemos la barra de herramientas la tendremos situada en la parte superior donde podremos crear carpeta y fichero y subir fichero.



[Figura 72: Barra herramientas]

Por otro lado, tenemos el menú con el botón derecho del ratón. En el podremos borrar y descargar ficheros.



[Figura 73: Menú ratón]

6.5.1 Crear carpeta

Para crear una carpeta nos situaremos en el directorio donde la queramos crear. Pulsaremos sobre el botón “Crear Carpeta” de la barra de herramientas.



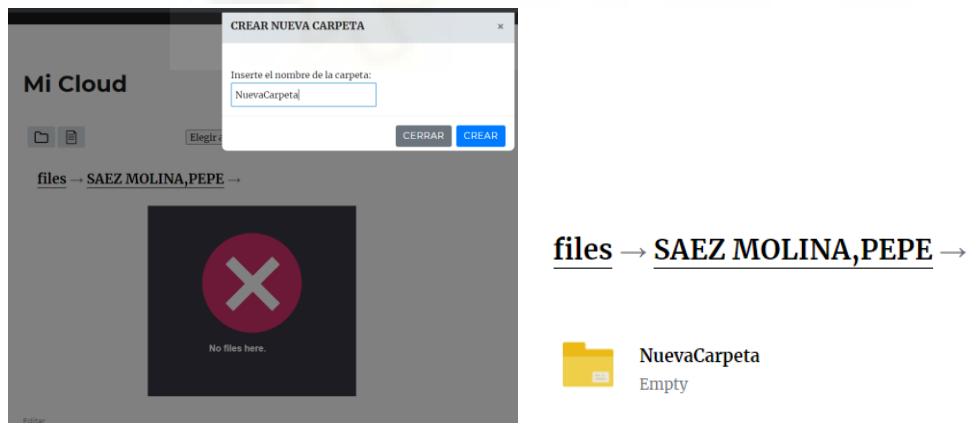
[Figura 74: Botón crear carpeta paso 1]

Se abrirá la siguiente pantalla para insertar el nombre y finalmente pulsar “Crear”.



[Figura 75: Popup crear nueva carpeta paso 2]

Por ejemplo, en la siguiente imagen se creará una carpeta en un directorio que no existe ninguna.



[Figura 76: Creación nueva carpeta paso 3]

La implementación la haremos conectándonos a DAV de NextCloud mediante un MKCOL de sable/dav.

```
$response = $client->request("MKCOL",$urlD);
```

[Figura 77: Creación nueva carpeta paso 4]

6.5.2 Crear fichero

Para crear un fichero nos situaremos en el directorio donde queramos crearlo. Pulsaremos sobre el botón “Crear Fichero” de la barra de herramientas.



[Figura 78: Creación nuevo fichero paso 1]

Nos aparecerá la siguiente pantalla, en ello deberemos insertar el nombre y el texto del fichero.



[Figura 79: Popup nuevo fichero paso 2]

Por defecto, todos los ficheros se guardarán con extensión .txt. No haría falta ponerla, en el caso de ponerla lo tomará como parte de nombre. Es decir, si cómo nombre se inserta erróneamente “Fichero.txt” el nombre del fichero se creará cómo “Ficherotxt.txt”.

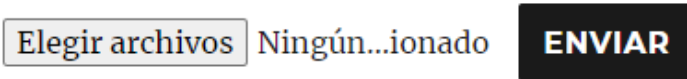
La implementación la haremos conectándonos a DAV de NextCloud mediante un PUT de sable/dav. Incluyendo cómo parámetros la ruta y el contenido del fichero.

```
$response = $client->request("PUT", $urlD, "".$contentNewFile."");
```

[Figura 80: Creación nuevo fichero paso 3]

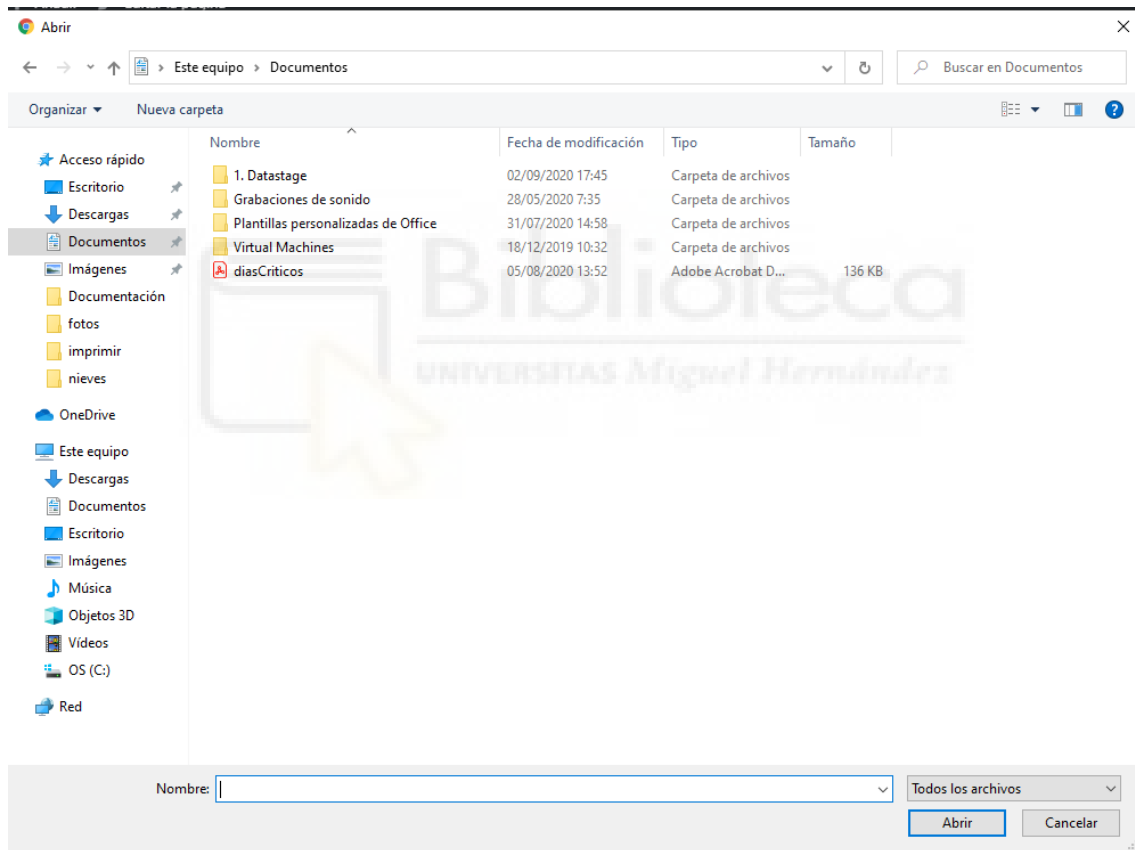
6.5.3 Subir fichero

Para subir el fichero nos situaremos en el directorio donde queremos crearlo. Pulsaremos sobre el “Elegir archivos” de la barra de herramientas.



[Figura 81: Menú subir fichero paso 1]

Se nos abrirá el explorador de archivos de Windows, seleccionaremos el fichero que queramos subir.



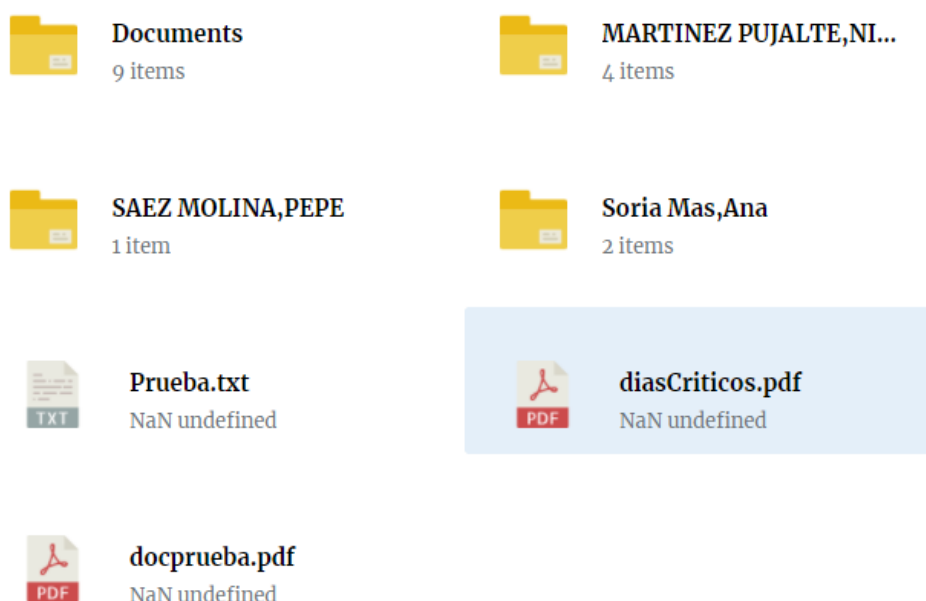
[Figura 82: Menú subir fichero 2]

Pulsaremos “Enviar”.



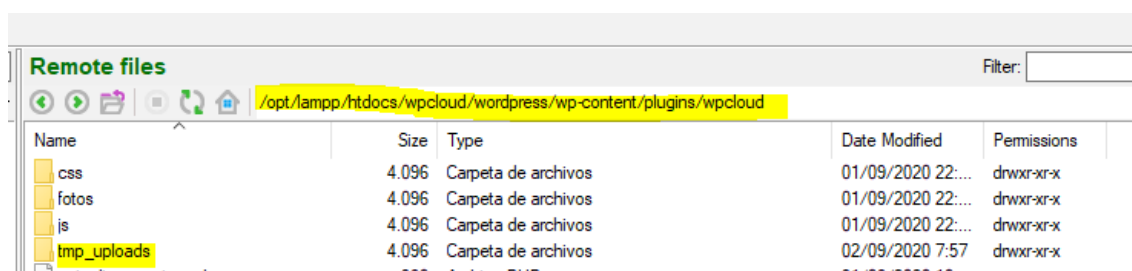
[Figura 83: Menú subir fichero 3]

Aparecerá en la carpeta correspondiente como se muestra en la siguiente imagen.



[Figura 84: Subir fichero paso 4]

Para realizar el software hemos necesitado crear una carpeta temporal denominada "tmp_uploads".



[Figura 85: Carpeta temporal paso 5]

Para implementar la subida del fichero deberemos utilizar la función de PHP ("Hypertext Preprocessor") `move_uploaded_file` que moverá el fichero del directorio a la carpeta temporal. Necesitará los parámetros con la ruta tanto del fichero de origen como la ruta destino.

Seguidamente se llama a la función que se ha creado *PutFicheroNextCloud*, con el nombre del fichero y la ruta de la carpeta que queremos guardar en NextCloud.

```
move_uploaded_file($temporal, $Destino);
PutFicheroNextCloud($NombreOriginal,$rutacarpeta);
```

[Figura 86: Subir archivo paso 6]

Utilizaremos las funciones para conectar y desconectar la sesión de curl. Para configurar la transferencia utilizaremos *curl_setopt_array()*. La dirección de la petición será a `"/nextcloud/remote.php/dav/files/"`. Los parámetros que enviaremos serán la dirección del fichero, las cabeceras, las credenciales, estado de la transferencia, método *put*, el tamaño esperado en bytes del fichero entre otros. Antes de finalizar la sesión, mediante *curl_exec* se ejecuta la sesión que se pasa por el parámetro.

```
$logfh = fopen("tmp_uploads/".$NombreFichero, 'r');
if ($logfh !== false) {
    print "Opened the log file without errors";
}

$params=array(
    'file' =>
        | '0' . realpath('/opt/lampp/htdocs/wpcloud/wordpress/wp-content/plugins/wpcloud/tmp_uploads/'.$NombreFichero)
);
echo str_replace('///','/',"localhost/nextcloud/remote.php/dav/files/".$User .'/'.$Urlfolder.'/'.$NombreFichero);
$defaults = array(
    CURLOPT_URL => str_replace('///','/',"localhost/nextcloud/remote.php/dav/files/".$User .'/'.$Urlfolder.'/'.$str_replace(" ", "%20", $NombreFichero)),
    CURLOPT_HTTPHEADER => array('OCS-APIRequest:true'),
    CURLOPT_USERPWD => $_COOKIE["cookieUser"]. ':' . $_COOKIE["cookiePassword"] . "1234",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_PUT => true,
    CURLOPT_INFILESIZE => filesize("/opt/lampp/htdocs/wpcloud/wordpress/wp-content/plugins/wpcloud/tmp_uploads/".$NombreFichero),
    CURLOPT_INFILE => $logfh,
    CURLOPT_VERBOSE => true,
    CURLOPT_POSTFIELDS => $params,
);
$ch = curl_init();
curl_setopt_array($ch, $defaults);
$resultput=curl_exec($ch);

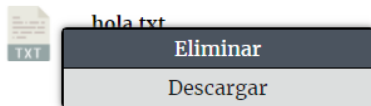
echo $resultput;

curl_close($ch);
fclose($logfh);
```

[Figura 87: Subir archivo paso 7]

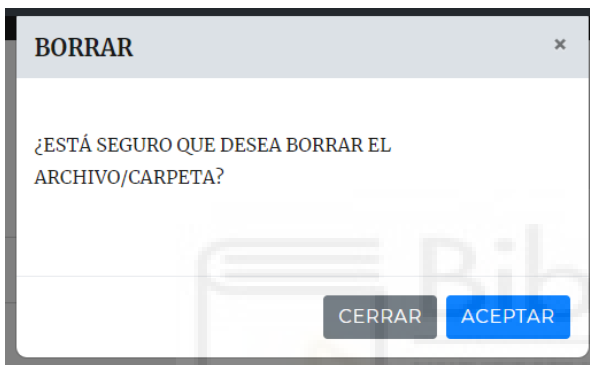
6.5.4 Borrar fichero

Para borrar los ficheros haremos clic con el botón derecho del ratón en el fichero y pulsaremos “Eliminar” para borrarlo.



[Figura 88: Menú borrar paso 1]

Nos aparecerá una ventana para confirmarlo y pulsaremos aceptar.



[Figura 89: Popup borrar paso 2]

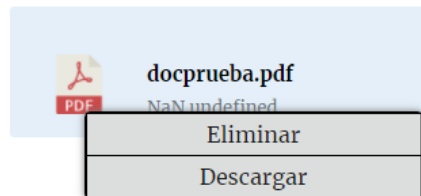
La implementación la haremos conectándonos a DAV de NextCloud mediante un DELETE de sable/dav.

```
$response = $client->request("DELETE", $urlD);
```

[Figura 90: Borrar fichero paso 3]

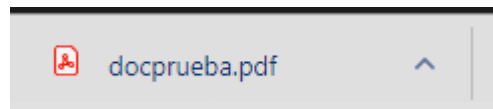
6.5.5 Descargar archivos

Para descargar archivos nos situaremos sobre él, haremos clic con el botón derecho y seleccionaremos descargar.



[Figura 91: Menú descargar archivo paso 1]

Comprobaremos cómo se descarga con el navegador.



[Figura 92: Descargar archivo paso 2]

A la hora de implementar el software utilizaremos curl. Inicializaremos y configuraremos la transferencia utilizando `curl_setopt_array()`. La dirección de la petición será a `"/nextcloud/remote.php/dav/files/"`. Los parámetros que enviaremos serán la dirección del fichero, las cabeceras, las credenciales, estado de la transferencia, subir fichero. Antes de finalizar la sesión, mediante `curl_exec` se ejecuta la sesión que se pasa por el parámetro.

```
$fp=fopen("tmp_uploads/".$nombrefich, "w+");
$defaults = array(
    CURLOPT_URL => str_replace("//", "/", 'localhost/nextcloud/remote.php/dav/files/'.$_COOKIE["cookieUser"]. $rutacloud),
    CURLOPT_HTTPHEADER => array('OCS-APIRequest:true', 'Content-Type: application/x-www-form-urlencoded'),
    CURLOPT_USERPWD => $_COOKIE["cookieUser"]. ':' . $_COOKIE["cookiePassword"] . $rcccloud,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_FILE
);
$ch = curl_init();
curl_setopt_array($ch, $defaults);
//curl_setopt($ch, $defaults, true);
$datafich = curl_exec($ch);
if($datafich === false){
    $resultcreateuser= array( 'message' => 'RO');
}else{
    //echo 'En teoria funciona';
    $resultcreateuser= array( 'message' => 'OR', 'namefile' => $nombrefich);
}
fwrite($fp, $datafich);
fclose($fp);
```

[Figura 93: Descargar archivo paso 3]

Una vez descargado a la carpeta temporal abrimos el fichero y definimos las cabeceras, leemos el fichero para cambiarle la ruta añadiendo el nombre del archivo.

```
$fileName=$_GET['filename'];
$fileName = basename($fileName);
$filePath = 'tmp_uploads/'.$fileName;
if(!empty($fileName) && file_exists($filePath)){
    // Define headers
    header("Cache-Control: public");
    header("Content-Description: File Transfer");
    header("Content-Disposition: attachment; filename=".str_replace(" ", "_",$fileName));
    header("Content-Type: application/zip");
    header("Content-Transfer-Encoding: binary");

    // Read the file
    readfile($filePath);

    unlink('tmp_uploads/'.$fileName);
    exit;
}else{
    echo 'The file does not exist.';
}
}
```

[Figura 94: Descargar archivo paso 4]

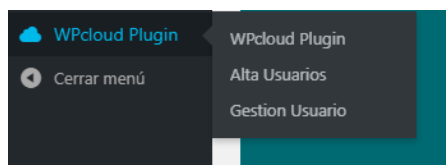
Para finalizar abriríamos la ruta mediante la función “window.open” para descargarlo.

```
var url = 'php_code_home_url(../)wp-content';
window.open(url, 'Descargar Fichero');
```

[Figura 95: Descargar archivo paso 5]

6.6 GESTIÓN USUARIO

Con esta herramienta vamos a poder administrar los permisos de los clientes. Este menú sólo será visible para los abogados del bufete y no estará habilitado para los clientes. Entraremos en el menú del plugin y seleccionar el submenú “Gestión Usuario”.



[Figura 96: Menú gestión usuario paso 1]

Nos aparecerá una tabla con los usuarios que hemos creado en MiCloud.

Listado de usuarios WPcloud							
Usuario	Carpeta	Permisos					Opciones
		Subir Ficheros	Descargar Ficheros	Crear Carpetas	Borrar Ficheros/carpeta	Crear Ficheros	
Nicasio33	Davo Davo,Nicasio	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="No"/>	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	Actualizar Eliminar
Josico	Pepe Reina,Josico	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="No"/>	<input type="text" value="No"/>	Actualizar Eliminar
tfgClient	tfgClient	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="No"/>	Actualizar Eliminar
tfgAna	Sorla Has,Ana	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="No"/>	<input type="text" value="Sí"/>	Actualizar Eliminar
pepe22	SAEZ MOLINA,PEPE	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="No"/>	<input type="text" value="No"/>	<input type="text" value="No"/>	Actualizar Eliminar

[Figura 97: Gestión usuario paso 2]

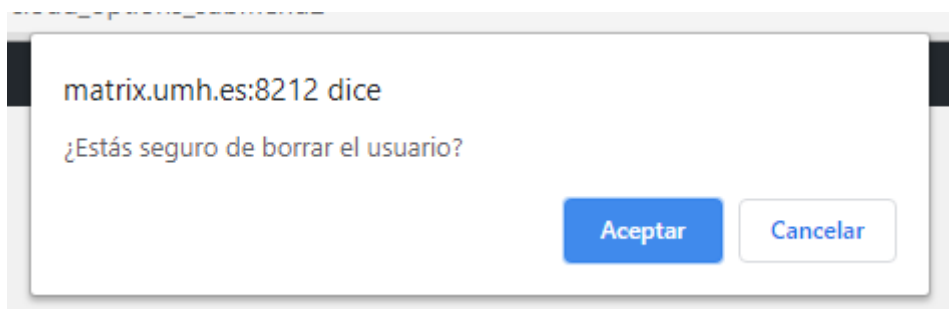
En esta tabla tendremos que seleccionar mediante el desplegable los permisos y pulsar sobre “Actualizar”. Los permisos que podemos deshabilitar son subir fichero, descargar archivo, crear carpeta, borrar ficheros, crear ficheros.

Por ejemplo si entráramos al usuario de pepe no podría cargar carpetas, y efectivamente, no tiene habilitado el botón de crear carpetas.



[Figura 98: Gestión usuario paso 3]

También tendremos la opción de pulsar sobre “Borrar” para eliminar el usuario. Nos aparecerá un mensaje de alerta.



[Figura 99: Gestión usuario paso 4]

Si pulsamos aceptar, la tabla se recargará y se borrará el usuario seleccionado.

Listado de usuarios WPcloud

Usuario	Carpeta	Permisos		
		Subir Ficheros	Descargar Ficheros	Crear Carpetas
Nicasio33	Davo Davo,Nicasio	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="No"/>
Josico	Pepe Reina,Josico	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>
tfgClient	tfgClient	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>
pepe22	SAEZ MOLINA,PEPE	<input type="text" value="Sí"/>	<input type="text" value="Sí"/>	<input type="text" value="No"/>

[Figura 100: Gestión usuario paso 5]

Para la implementación tendremos que actualizar los datos en la tabla de base de datos que creamos cuando los usuarios se dan de alta. Para ello realizaremos un “insert”. Como se muestra en la imagen.

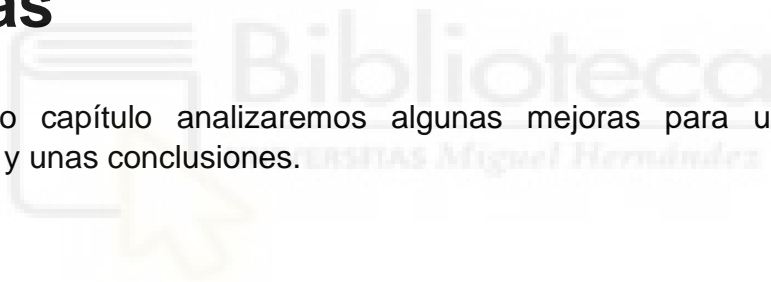
```
$sqlinsertcloud = "Update wpcloud set subirfich='". $boolsubida .  
"', bajafich='". $boolbajada ."', crearsubcarpetas='". $boolcarpeta .  
"', crearfichero='". $boolcrear ."', borrarelemento='". $boolborrar .  
"' where id='". $idcloud ."'";  
$updatewpcloud = $conexionbd -> query($sqlinsertcloud);  
  
$conexionbd -> close();
```

[Figura 101: Gestión usuario paso 6]

En el caso de tener la opción “NO” mediante el elemento “display” y la propiedad “none” de CSS (Cascading Style Sheets) ocultaremos los botones para que tenga deshabilitado las herramientas correspondientes.

Capítulo 7: Conclusiones y Líneas Futuras

En el último capítulo analizaremos algunas mejoras para un futuro, el presupuesto y unas conclusiones.



7.1 Líneas futuras

El proyecto desarrollado y expuesto está en sus inicios. El desarrollo pretende conseguir un diseño básico pero eficaz sobre el almacenamiento de archivos.

Varias de las mejoras futuras podrían ser las siguientes:

- Mejorar la usabilidad de subida de ficheros mediante “Drag and Drop”, es decir, arrastrando y soltando los ficheros.
- Mejorar el diseño del plugin para que se adapte a las diferentes plantillas existentes en WordPress. Añadiendo un menú de personalización.
- Añadir otras herramientas. Como podría ser, botón de compartir, mover o copiar.
- Mejorar el diseño de gestión de usuarios pudiendo cambiar cualquier dato personal de los usuarios. Como, por ejemplo, cambiar el correo electrónico.
- Añadir funcionalidad de descarga de carpetas comprimiéndolas.
- Incluir un reproductor de vídeo, audio o visor de archivos Office.
- Estudiar más medidas de seguridad.
- Barajar la posibilidad de comercialización en la propia tienda online de WordPress. Ya que el plugin pudiera ser utilizado para otro tipo de sectores.
- Realizar la implementación de una aplicación móvil.

7.2 Conclusiones

Una vez concluido el desarrollo del proyecto se han evaluado los resultados conseguidos a lo largo de mismo.

El objetivo principal del proyecto era el desarrollar un plugin en WordPress que permitiera gestionar a los bufetes de abogados el almacenamiento de archivos de forma “segura”. Analizando la mejor forma para dar una solución hemos llegado a la conclusión que era teniendo un gestor de archivos en su propio servidor. Después de ver todas las opciones de software libre que tenemos en el mercado, hemos pensado que la mejor opción es NextCloud. Debido a que implementa muchas medidas de seguridad, tiene pocas limitaciones en sus licencias de contratación, compatibilidad con archivos office gratuitamente, cualquiera de sus versiones tiene licencias de software libre, el precio y la accesibilidad a la interfaz mediante API (“Application Programming Interfaces”).

Por otro lado, hemos visto que la mayoría de pequeñas y medianas empresas, a las cuales va orientado nuestro proyecto, no tienen un gran conocimiento sobre programación de páginas web y utilizan un gestor de contenido para crearlas. Entre todos los gestores de contenido más usados hemos optado por escoger WordPress. Ya que, el software es libre, recibe muchas visitas, está en continua actualización y además es fácil de usar. Finalmente hemos visto la manera de implementar en él el gestor de archivos NextCloud.

Podemos concluir diciendo que dichos objetivos han sido cumplidos según las expectativas planteadas al comienzo del proyecto.

7.3 Presupuesto

Para la realización de un presupuesto de comercialización habría que tener en cuenta los tiempos de investigación, los costes de comercialización, los costes de licencias o certificados necesarios. Sin embargo, para este estudio teniendo en cuenta que todas las herramientas son de software libre, no tenemos coste de licenciamiento. Sólo vamos a tener en cuenta los costes de instalación y configuración del servidor. A lo que habría que añadir las horas de desarrollo en caso de ser un proyecto Ad-hoc. En nuestro caso los costes de instalación y configuración del servidor han sido subsanados por la universidad por tanto no ha habido ningún costo.

Bibliografía

Introducción:

- [1]: <https://www.powerdata.es/transformacion-digital>
- [2]: <https://es.digitaltrends.com/tendencias/que-es-el-internet-de-las-cosas/>
- [3]: <https://revistaingenieria.deusto.es/tag/iot/>
- [4]: <https://www.abogacia.es/wp-content/uploads/2018/02/ABOGACIA-Cloud-computing.pdf>
- [5]: <https://destinonegocio.com/pe/negocio-por-internet-pe-pe/que-ventajas-ofrece-migrar-a-la-nube-para-mi-empresa/>
- [6]: <https://dataprius.com/almacenamiento-nube-cloud>

Cloud Computing:

- [7]: <https://www.muycomputerpro.com/2018/07/03/historia-cloud-computing>
- [8]: <https://www.managementsolutions.com/sites/default/files/publicaciones/es/La-nube.pdf>
- [9]: <https://www.salesforce.com/mx/cloud-computing/>
- [10]: <https://www.cisga.es/conoces-origen-del-cloud-computing-nube-informatica/>
- [11]: <https://www.webempresa.com/hosting/diferencias-hosting-compartido-y-dedicado.html>
- [12]: <https://www.comparahosting.com/hosting/>
- [13]: <https://azure.microsoft.com/es-es/overview/what-is-saas/>
- [14]: <https://azure.microsoft.com/es-es/overview/what-is-paas/>

- [15]: <https://azure.microsoft.com/es-es/overview/what-is-iaas/>
- [16]: <https://azure.microsoft.com/es-es/overview/what-are-private-public-hybrid-clouds/>
- [17]: <https://www.adslzone.net/reportajes/internet/comparativa-almacenamiento-nube/>
- [18]: <https://www.xataka.com/basics/google-one-dropbox-onedrive-icloud-todas-opciones-frente-a-frente>

Herramientas

- [19]: <https://nextcloud.com/support/>
- [20]: <https://www.tecmint.com/free-open-source-cloud-storage-tools-for-linux/>
- [21]: <https://blog.irontec.com/mi-experiencia-con-owncloud/>
- [22]: <http://www.nethserver.org/nextcloud-easy-installmanageuse-1click/>
- [23]: <https://www.1and1.es/digitalguide/servidores/know-how/alternativas-a-owncloud-existen-otras-soluciones/>
- [24]: <http://luengocabanillas.com/2016/04/23/pydio-vs-owncloud-vs-open365/>
- [25]: <https://www.hostinger.es/tutoriales/tutorial-wordpress-plugins-crear-plugin-wordpress/#gref>
- [26]: <https://es.wordpress.org/>
- [27]: https://codex.wordpress.org/es:Escribiendo_un_Plugin
- [28]: <https://nonopp.com/cursos/WordPress/Manual-WordPress-CursoCamara2.pdf>

[29]: <https://www.qed42.com/blog/using-curl-commands-webdav>

[30]: <https://www.cloudwards.net/what-is-webdav/>

Seguridad

[31]: <https://www.arturogarcia.com/medidas-seguridad-wordpress/>

[32]: <http://aulacm.com/seguridad-y-proteger-wordpress/>

[33]: <http://www.wpbeginner.com/wordpress-security/>

[34]: <https://premium.wpmudev.org/blog/wordpress-security-tips/>

[35]: <https://www.corephp.com/blog/15-advanced-security-tips-to-make-your-wordpress-site-bulletproof/>

[36]: <https://www.codeinwp.com/blog/secure-your-wordpress-website/>

[37]: <https://wpdistrict.sitelock.com/wp-content/uploads/pdfs/getting-started-in-wordpress-security.pdf>

[38]: <https://clasesdeseguridadinformatica.files.wordpress.com/2014/03/administrador-de-servidores.pdf>

[39]: <http://seguridadidga.blogspot.com/2015/11/ssl-y-tsl.html>

[40]: <https://www.hostinger.es/tutoriales/ssl-tls-https/>

[41]: <https://nextcloud.com/secure/>

Solución Integrada y Plugin WPCloud

- [42]: https://docs.nextcloud.com/server/latest/developer_manual/client_apis/OCS/ocs-share-api.html
- [43]: <https://sabre.io/dav/locks/>
- [44]: https://www.w3schools.com/js/js_json_intro.asp
- [45]: <http://deletesql.com/>
- [46]: <http://deckerix.com/blog/como-crear-usuarios-de-wordpress-desde-php/>
- [47]: <https://fernando-gaitan.com.ar/aprendiendo-php-parte-19-subir-archivos-al-servidor/>
- [48]: <https://www.hostinger.es/tutoriales/que-es-un-hosting>
- [49]: <https://www.hostinger.es/tutoriales/comando-curl/>
- [50]: <https://blog.educacionit.com/2017/08/16/guia-sobre-como-utilizar-curl-en-php/>
- [51]: <https://www.php.net/manual/es/function.curl-setopt.php>
- [52]: <https://developer.wordpress.org/reference/functions/>

Bibliografía de imágenes

[Figura 1]:

<https://sophimania.pe/tecnologia/internet-tics-y-startups/en-2020-habra-25-mil-millones-de-dispositivos-conectados-por-el-internet-de-las-cosas-iot/>

[Figura 2]:

<https://blog.sysfore.com/use-cloud-computing-to-transform-your-startup-into-tomorrows-mid-market-powerhouse/>

[Figura 3]:

<https://www.neoteo.com/murio-john-mccarthy-referente-de-la-inteligencia-a/>

<http://www.tecnologiahechapalabra.com/datos/eventos/articulo.asp?i=7552>

[Figura 4]: Montaje fotos logo imágenes Google.

[Figura 5]:

<https://www.webempresa.com/hosting/diferencias-hosting-compartido-y-dedicado.html>

[Figura 6]:

<https://www.codematters.online/demystifying-azure-paas/cloud-computing-pyramid-iaas-paas-saas/>

[Figura 7]:

<http://altcomp.mx/cuales-son-las-diferencias-entre-nube-hibrida-privada-y-publica/>

[Figura 8]: Montaje fotos logo imágenes Google.

[Figura 9] Icono escritorio propio ordenador.

[Figura 10]: Icono escritorio propio ordenador.

[Figura 11]: Montaje fotos logo imágenes Google.

[Figura 12]: captura ordenador al propio WordPress.

[Figura 13]: captura ordenador al propio WordPress.

[Figura 14]:

<https://es.wordpress.org/>

[Figura 15]:

<https://www.apachefriends.org/es/index.html>

[Figura 16 a la Figura 33]: captura ordenador propio.

[Figura 32 a la Figura 38]:

<https://nextcloud.com/>

[Figura 39 a la 101]: captura ordenador propio.