



IES Gran Capitán

Departamento de Informática

Ciclo Formativo de Grado Superior de
Administración de Sistemas Informáticos

Módulo de Proyecto Integrado

Miguel Ángel Coletto López – 2ºA.S.I.R

David Toro Martínez – 2ºA.S.I.R

Proyecto: ownCloud para Directivos

Curso <2019/2020>

Contenido

1 Introducción.....	3
2 Objetivos y requisitos del Proyecto.....	4
3 Estudio previo.....	4
3.1 Estudio de posibles soluciones.....	4
3.2 Elección de la mejor solución.....	5
4 Plan de trabajo.....	6
5 Diseño.....	8
5.1 Esquema de red.....	8
5.2 Recursos.....	8
5.3 Estructura organizativa de usuarios, grupos y carpetas.....	8
5.4 Base de datos.....	9
6 Implantación.....	10
6.1 Preparación del entorno.....	10
6.1.1 Instalación Básica.....	10
6.1.2 Creación de usuarios y grupos.....	11
6.1.3 Cifrado y SSL.....	12
6.1.4 Compartir una carpeta del servidor local.....	13
6.1.5 Alta disponibilidad.....	14
6.1.6 Edición de documentos office con Collabora Online.....	16
6.2 Configuración del servidor de correo.....	17
6.3 Incorporar los directorios del NAS a la nube.....	17
6.4 Blindaje y mejoras del servidor.....	19
6.5 Alta disponibilidad con KeepAlived.....	21
6.6 Visor de PDF y Editor de .TXT.....	22
7 Personal y Presupuesto.....	22
8 Grado de consecución de objetivos.....	22
9 Competencias aplicadas.....	24
10 Problemas encontrados.....	24
11 Futuras mejoras.....	27
12 Bibliografía.....	27

1 Introducción

Este proyecto surge a partir de la necesidad de tener un espacio virtual online privado, seguro y confiable donde la dirección del instituto I.E.S Gran Capitán (y/o otros departamentos aparte de la dirección) pueda almacenar archivos y documentos para trabajar con ellos. Se hace especial hincapié en que se debe establecer un control de acceso estricto dada la sensibilidad de la información que pueden contener estos ficheros pero también debe existir cierta flexibilidad de trabajo para aquellos archivos que deben ser compartidos entre distintas personas de dirección y/o otros departamentos.

El mayor problema existente, es que al estar estos datos almacenados (ya sea información sensible o protegida) en sistemas como Google Drive es que no se está cumpliendo la ley de protección de datos (RGPD). Los datos deben estar almacenados en un país del Espacio Económico Europeo o un país que ofrezca un nivel de protección equivalente (que haya sido así acordado por la Agencia Española de Protección de Datos o por Decisión de la Comisión Europea).

Para los casos de tratamientos especiales de datos personales que puedan suponer un mayor riesgo el responsable debe obtener el consentimiento expreso de los alumnos (si son mayores de 14 años) o de los padres o tutores (si son menores de 14 años) para aplicar dicho tratamiento a las imágenes con fines de identificación, y asegurarse que esta tecnología se utiliza únicamente para fines concretos especificados y legítimos.

La responsabilidad del cumplimiento de las medidas de seguridad debe entenderse siempre compartida entre los diferentes actores intervinientes (responsable de la aplicación, Centro Educativo y usuarios), debiendo en todo caso el responsable de la aplicación facilitar las medidas técnicas adecuadas para garantizar la seguridad de los datos tratados, y el Centro aplicarlas o utilizarlas correctamente, además de implementar las medidas organizativas apropiadas. Así, por ejemplo, la aplicación debe proveer mecanismos que permitan la realización de copias de seguridad o la descarga de los datos, de tal forma que el Centro pueda cumplir con las obligaciones que le son exigibles al respecto, introduciendo en su política de seguridad la realización de copias de seguridad de los datos tratados mediante estas aplicaciones, y realizando efectivamente la realización de dichas copias.

El RGPD estipula requisitos concretos que deben cumplir todas las empresas y organizaciones que tengan su sede en Europa o que presten servicios a usuarios europeos:

- Regula cómo pueden recoger, usar y almacenar datos personales las empresas.
- Desarrolla los requisitos actuales de documentación y creación de informes para aumentar la responsabilidad proactiva.
- Autoriza que se impongan multas a las empresas que no cumplan dichos requisitos.

En definitiva, por esto mismo nos urge buscar una solución y emigrar los datos a Owncloud, ya que no se nos asegura que estos datos almacenados en Google Drive, sean tratados acorde a la ley de

protección de datos impuesta en la Unión Europea.

2 Objetivos y requisitos del Proyecto

Nuestro objetivo es montar un servicio de almacenamiento en nube. Esta solución cubre las necesidades explicadas en la introducción.

En concreto necesitamos:

1. Instalar un software de nube privada accesible desde la web
2. Configurar el servidor web para que sea seguro (HTTPS)
3. Gestionar las cuentas de la nube de forma que cumplan los requisitos de seguridad de la RGPD
4. Implementar mecanismos de alta disponibilidad
5. Cubrir posibles agujeros de seguridad

Adicionalmente se debe de configurar este servicio de nube de forma que cubra las necesidades de control de acceso y sea posible extender las funcionalidades del servicio para cubrir cualquier necesidad adicional que el I.E.S Gran Capitán requiera.

3 Estudio previo

3.1 Estudio de posibles soluciones

La solución que más se ajusta a las necesidades es una nube privada propia alojada en los servidores del instituto, ajustándonos con esto a la Ley Orgánica de Protección de Datos.

OwnCloud es un software popular de almacenamiento en nube y fue propuesto de forma inicial para este proyecto pero existen alternativas. Hemos estudiado ,además de ownCloud, NextCloud, Seafile, Syncthing, SpiderOak, SparkleShare, Resilio Sync y Goodsync.

Resilio Sync y Goodsync los hemos descartado inicialmente al ser software de pago. Goodsync es presentado como software gratuito con una opción de pago para mejorar sus características pero en realidad la version gratuita es en realidad una versión de prueba ya que limita incluso el numero de ficheros que puedes almacenar en el servidor a 100. SpiderOak fue descartado por ser un software de codigo cerrado, menos popular y que presenta tiempos de carga lentos.

Con el resto de opciones hemos elaborado una tabla con sus ventajas e inconvenientes frente a Owncloud

<u>Software</u>	<u>Ventajas</u>	<u>Incovenientes</u>
Nextcloud	-Protección contra ataques de fuerza bruta -Interfaz personalizable	-Añadir funciones propicia el aumento de fallos y errores (mala gestión de modulos y

	-Permite conversaciones de audio y video	plugins) -Actualizar puede ocasionar problemas
Syncthing	-Rápido y eficiente - Instalar y actualizar es fácil y sencillo	-No utiliza un sistema de autenticación basado en usuarios y contraseñas sino en certificados de dispositivos lo cual lo hace muy poco práctico para este proyecto en cuestión
SparkleShare	-Historial de versiones de archivos que permite revertir a versiones anteriores y mantener un historial de cambios	-Contiene limitaciones sobre que se puede almacenar (no se puede usar git ya que está basado en git) -Poco eficiente - El menos intuitivo de usar de todos desde el punto de vista de un usuario
Seafile	-Interfaz de usuario limpia y simple -Incorpora mensajería, comentarios sobre ficheros, contactos y una wiki	-Aunque su versión gratuita es bastante completa se vuelve insuficiente cuando necesitamos escalabilidad o la nube se vuelve muy grande
OwnCloud (Respecto al resto)	-A priori una nube bastante básica pero puede extender sus funcionalidades con módulos y plugins según se necesite -Permite montar otros sistemas de almacenamiento y nubes de terceros	-El cifrado se realiza del lado del servidor y no existe cifrado p2p cuando se comparten archivos

3.2 Elección de la mejor solución

Las razones por las que pensamos que ownCloud es la mejor alternativa para este proyecto son:

1. Es muy flexible y se le pueden ir incorporando nuevas funcionalidades fácilmente.
2. Es software libre y gratuito.
3. Es rápido y seguro.
4. Intuitivo de usar para los usuarios.
5. Permite centralizar otros servicios de almacenamiento. En el caso de este proyecto se pueden montar carpetas en la intranet y ftp.

Owncloud empezó a desarrollarse en enero de 2010 para competir contra las nubes propietarias existentes en el mercado.

Se fundó la compañía Owncloud Inc. en 2011 y el código de Owncloud se trasladó a GitHub. La compañía fue fundada por Markus Rex, Holger Dyroff y Frank Karlitschek.

El software está escrito en PHP y JavaScript ya que es una aplicación web, todo su Código es

abierto y libre, está en github.com/owncloud y su licencia está bajo una GNU AGPLv3

4 Plan de trabajo

Primera fase 18/03 – 25/03

Tarea	Duración (en horas)	Personal
Estudio de soluciones existentes	1	David
Elección de la opción más adecuada al entorno del cliente	2 cada uno	David y Miguel Ángel
Investigación en profundidad de la opción elegida (recursos hw y sw)	2 cada uno	David y Miguel Ángel
Preparación de máquinas de trabajo(máquinas virtuales)	1,5	David y Miguel Ángel

Segunda fase 01/04 – 08/04

Tarea	Duración (en horas)	Personal
Búsqueda de información actualizada	1	David
Implantación de servidor básico en Debian(fallido al no poder resolver dependencias)	4	Miguel Ángel
Implantación de servidor básico en Ubuntu Server 18.04.4(exitosa)	1,5	David y Miguel Ángel
Implantación de tráfico seguro HTTPS	1	Miguel Ángel
Montar carpeta en local como almacenamiento externo(solo lectura)	2	Miguel Ángel
Creación de usuarios y grupos de prueba	0,5	David

Tercera fase 01/04 - 08/04

Tarea	Duración(en horas)	
Programación de la copia de seguridad(archivos de usuario, configuración y base de datos)	0,5	Miguel Ángel
Cifrado de archivos(investigación e implementación)	0,5	Miguel Ángel
Script de implementación automático en Júpiter(migración)	1,5	David
Dominio + Servidor de correo	2,5	David

Cuarta fase 17/04 – 24/04

Tarea	Duración (en horas)	Personal
Instalación de un servidor de correo asociado a owncloud para la administración segura de contraseñas	0,5	Miguel Ángel
Elaboración de la estructura de usuarios y carpetas	1,5	Miguel Ángel
Investigación logueo con las cuentas de google	1	Miguel Ángel
Implantación de un servidor de colabora online para la edición de documentos office (problema)	1,5	Miguel Ángel

Quinta fase 25/04 – 09/05

Tareas	Duración	Personal
Alta disponibilidad	4	Miguel y David

Sexta fase 10/05 – 25/04

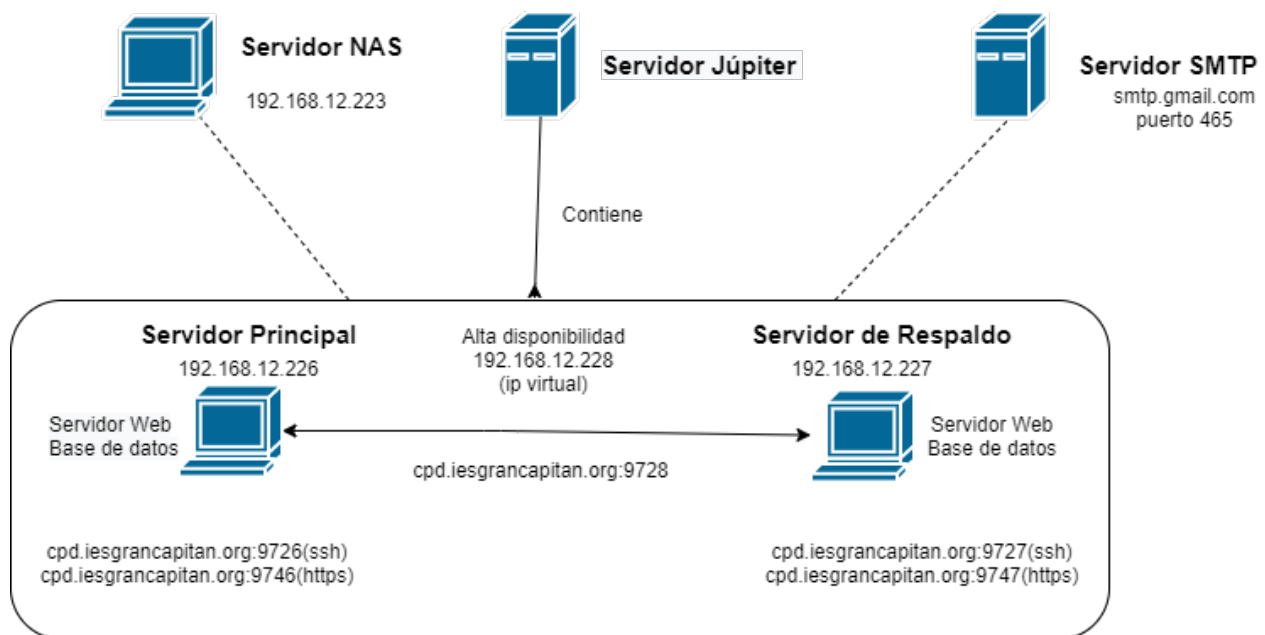
Tareas	Duración	Personal
Implantación en Júpiter del servidor Owncloud básico	3	Miguel y David
Alta disponibilidad básica	2	Miguel y David
Https/Cifrado/Correo electrónico	2	Miguel y David
Investigación sobre posibles mejoras(OAUTH2 Y LDAP)	0,5	Miguel y David

Séptima fase 26/05 – 18/06

Integración del NAS	0,5	Miguel
Creación de usuarios y grupos(con sus permisos)	1	David
Estructura de carpetas	1	David
Alta disponibilidad completa	8	Miguel y David
Esquema de Red	0,5	David
Finalizar Documentación	12	Miguel y David

5 Diseño

5.1 Esquema de red



5.2 Recursos

Siguiendo el esquema anterior la maquina del Servidor IES Gran Capitan es un NAS QNAP Modelo TS-469L

Los 2 servidores ownCloud tanto el principal como de respaldo son máquinas virtuales con Ubuntu Server 18.04, 40 GB de disco duro y 2 GB de RAM cada una.

5.3 Estructura organizativa de usuarios, grupos y carpetas

Los usuarios y grupos los organizaremos por grupos. Los grupos planteados necesarios serían los siguientes: DACE, SECRETARIO, JEFATURA, VICEDIRECCION, DIRECCION, ADMON

Los estructuraremos de la siguiente manera:

DACE: Juan Rivera y Rosa

SECRETARIA: Joaquín Mesa

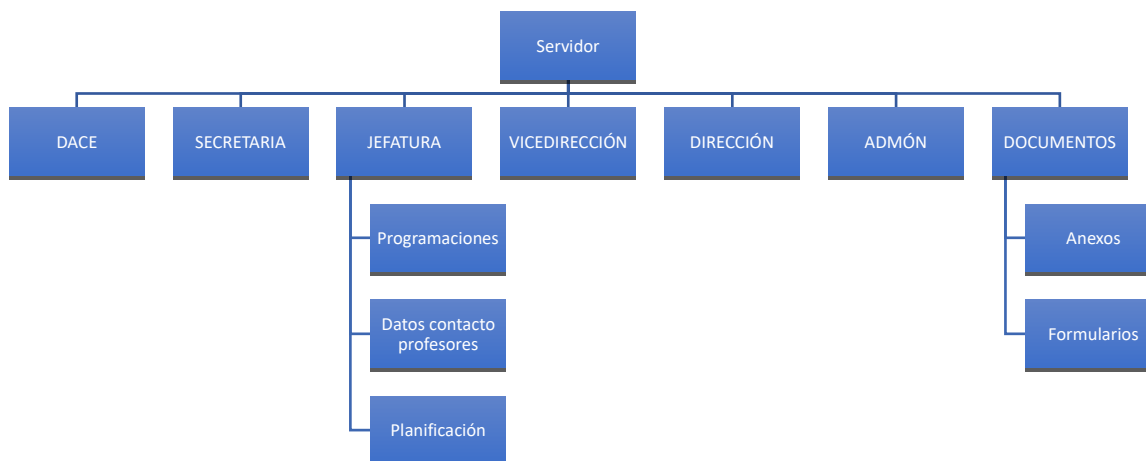
JEFATURA: Raúl Márquez, Ana Hernández y Jorge

VICEDIRECCIÓN: Juan Rivera

DIRECCIÓN: Carmen Domingo

ADMON: Raúl Márquez y Carmen Domingo

Por cada grupo que exista, debemos crear la carpeta correspondiente. Además habrá una carpeta documentos que contendrá anexos y formularios. La estructura es la siguiente:



Cada grupo tiene su subcarpeta del mismo nombre: DOC_[nombreGrupo], para que cuando haya más de un usuario en el grupo puedan compartir la misma

5.4 Base de datos

La base de datos de ownCloud es bastante compleja, contiene mas de 30 tablas por lo que no vamos a entrar a explicar cada tabla y campo pero si algunas tablas que contienen información útil que puede ser interesante para un administrador de sistemas.

oc_groups: almacena los grupos

oc_users: almacena el uid y la contraseña (encriptada) de los usuarios

oc_accounts: almacena información relacionada con las cuentas de los usuarios como el correo, su id, la cuota, su última fecha de conexión o su directorio donde almacena los archivos

oc_files_trash: muestra los archivos que han sido borrados junto al path donde estaban

oc_filecache: contiene información relacionada de todos los archivos y carpetas que tiene ownCloud

Ejemplo de una consulta en oc_filecache

```
mysql> select fileid,storage,path from oc_filecache;
```

fileid	storage	path
1	1	cache
2	1	files
3	1	files/ownCloud Manual.pdf
4	1	files/Documents
5	1	files/Documents/Example.odt
6	1	files/Photos
7	1	files/Photos/Squirrel.jpg
8	1	files/Photos/San Francisco.jpg
9	1	files/Photos/Paris.jpg
10	1	files/Photos/Paris.jpg
11	2	avatars
12	2	files_external
13	1	files_trashbin/files/proyecto documentacion fase 3.odt.d1590680265
14	2	files_encryption
15	2	files_encryption/OC_DEFAULT_MODULE
16	2	files_encryption/OC_DEFAULT_MODULE/pubShare_d74718fb.publicKey
17	2	files_encryption/OC_DEFAULT_MODULE/pubShare_d74718fb.privateKey
18	2	files_encryption/OC_DEFAULT_MODULE/master_d74718fb.publicKey
19	2	files_encryption/OC_DEFAULT_MODULE/master_d74718fb.privateKey
20	2	files_encryption/OC_DEFAULT_MODULE/master_d74718fb.privateKey
21	3	cache
22	3	files
23	3	files/ownCloud Manual.pdf
24	3	files_encryption
25	3	files_encryption/keys
26	3	files_encryption/keys/files
27	3	files_encryption/keys/files/ownCloud Manual.pdf
28	3	files_encryption/keys/files/ownCloud Manual.pdf/OC_DEFAULT_MODULE
29	3	files_encryption/keys/files/ownCloud Manual.pdf/OC_DEFAULT_MODULE/fileKey
30	3	files_encryption/keys/files/ownCloud Manual.pdf/OC_DEFAULT_MODULE/master_d74718fb.shareKey
31	3	files_encryption/keys/files/ownCloud Manual.pdf/OC_DEFAULT_MODULE/master_d74718fb.shareKey
32	3	files/Documents

6 Implantación

Para instalar ownCloud hemos preparado una pila LAMP 7.2 y ownCloud versión 10.4.1. OwnCloud necesita mínimo 128 MB de RAM y se recomienda un mínimo de 512 MB. Lo ideal es 1GB ya que queremos implementar funciones extras como el cifrado de datos.

6.1 Preparación del entorno

6.1.1 Instalación Básica

En nuestro caso, hemos descargado primeramente Ubuntu Server 18.0.4.1 desde su página oficial y lo usaremos como entorno para el servidor de ownCloud.

Una vez instalado será necesario añadir los repositorios de php versión 7.0 (ya que con php 7.2 puede dar problemas)

Será necesario instalar apache y su módulo de php libapache2-mod-php7.0 además de los siguientes módulos para php:

```
php7.0-gd          php7.0-json    php7.0-mysql
php7.0-curl        php7.0-intl    php7.0-mcrypt
php7.0-imagick     php7.0-zip     php7.0-xml
php7.0-mbstring
```

Hemos instalado también mysql-server para alojar nuestra base de datos de owncloud.

Por supuesto es necesario descargar el propio software que vamos a instalar, ownCloud, hemos descargado la versión 10.04 de owncloud desde su página web.

Lo hemos descomprimido en /var/www/html. Debemos darle permisos recursivos a la carpeta entera a apache (www-data)

Ya que vamos a alojar un sitio web hemos creado el archivo de configuración del sitio y le hemos

hecho un enlace simbólico a la carpeta sites-enabled

Para el correcto funcionamiento de ownCloud se deben de habilitar los siguientes módulos de apache (algunos pueden aparecer ya habilitados): env, dir, mime, rewrite y headers

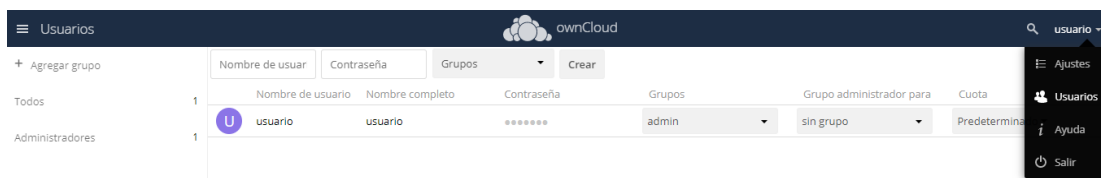
Una vez hecho esto reiniciamos el servicio de apache

Por último se debe de crear la base de datos y un usuario dentro de la base de datos el cuál usaremos para poder escribir en ella durante la instalación de owncloud(no es adecuado usar usuario root)

Una vez hecho esto se accede desde una máquina con entorno gráfico a <http://ip-servidor/owncloud> y procedemos a su instalación (similar a la instalación de un CMS como WordPress).

6.1.2 Creación de usuarios y grupos

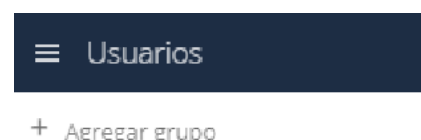
Una vez dentro del usuario creado durante la instalación, pinchando sobre nuestro usuario y haciendo clic en usuarios se nos despliega la siguiente pantalla.



Ahora para crear usuarios bastaría con introducir un nombre de usuario y una contraseña en la parte superior, además de al grupo que pertenece. Una vez implementemos SSL y cifrado no sera posible poner contraseñas a mano, los usuarios se crearan mediante nombre y correo electrónico y con un token único ellos asignaran su contraseña.

<input type="text" value="direccion2"/>	<input type="password" value="*****"/>	<input type="text" value="admin"/>	<input type="button" value="Crear"/>
---	--	------------------------------------	--------------------------------------

Ahora para crear un grupo, bastaría con pinchar a arriba a la izquierda en la misma pestaña en "Agregar grupos"



Ahora al darle click, nos aparece un cuadro de texto e introduciremos el nombre del grupo

<input type="text" value="Direccion"/>	<input type="button" value="+"/>
--	----------------------------------

Ahora, ya nos deja añadir dichos usuarios al grupo "Direccion"

	Nombre de usuario	Nombre completo	Contraseña	Grupos
D	direccion	direccion	••••••••	admin
D	direccion2	direccion2	••••••••	<input checked="" type="checkbox"/> admin
U	usuario	usuario	••••••••	<input type="checkbox"/> Direccion

También nos permite indicar el usuario de qué grupo será administrador, por ejemplo el usuario dirección sera el administrador del grupo dirección.

	Nombre de usuario	Nombre completo	Contraseña	Grupos	Grupo administrador para
D	direccion	direccion	••••••	admin	Direccion
D	direccion2	direccion2	••••••	admin	<input checked="" type="checkbox"/> Direccion
U	usuario	usuario	••••••	admin	sin grupo

Lo cual le permitirá administrar dicho grupo.

Por último también se nos permite indicar la cuota de los usuarios(el espacio máximo que podrán usar)

Nombre de usar		Contraseña		admin	Crear	
	Nombre de usuario	Nombre completo	Contraseña	Grupos	Grupo administrador para	Cuota
D	direccion	direccion	••••••	admin	Direccion	Predeterminado
D	direccion2	direccion2	••••••	admin	sin grupo	<div>Predeterminado ilimitado 1 GB 5 GB 10 GB Otro ...</div>
U	usuario	usuario	••••••	admin	sin grupo	

6.1.3 Cifrado y SSL

Lo primero que hemos implementado con respecto a la seguridad es el acceso por https en lugar de http y el cifrado de los datos de los usuarios. Debido a que nos encontramos en un entorno de pruebas local hemos usado un certificado autofirmado con la herramienta openssl.

Para habilitar el cifrado de datos en ownCloud debe de habilitarse el módulo de cifrado y el administrador debe indicar al servidor que cifre los datos de forma automática. Entrando desde un navegador en nuestro servidor ownCloud con el usuario administrador podemos hacer todo esto. En la pestaña de ajustes > administración > aplicaciones. Podemos habilitar el modulo

Ajustes

Personal





- General
- Almacenamiento
- Compartiendo
- Seguridad
- Cifrado
- Adicional

Administración

- Aplicaciones
- General
- Almacenamiento
- Cifrado
- Compartiendo
- Ayuda & Trucos

Gestión de aplicaciones

Mostrar aplicaciones desactivadas

 <p>Admin Config Report 0.2.0 por owncloud.org (Licencia AGPL)</p> <div style="border: 1px solid green; padding: 2px; display: inline-block;">✓ Oficial</div> <p>Mostrar descripción...</p> <p>Desactivar</p>	 <p>Default encryption module 1.4.0 por Bjoern Schiesle, Clark Tomlinson (Licencia AGPL)</p> <div style="border: 1px solid green; padding: 2px; display: inline-block;">✓ Oficial</div> <p>Mostrar descripción...</p> <p>Desactivar</p>
 <p>Share Files 0.12.0 por Michael Gapczynski, Bjoern Schiesle (Licencia AGPL)</p> <div style="border: 1px solid green; padding: 2px; display: inline-block;">✓ Oficial</div> <p>Mostrar descripción...</p> <p>Desactivar</p>	 <p>Update notification 0.2.1 por Lukas Reschke (Licencia AGPL)</p> <div style="border: 1px solid green; padding: 2px; display: inline-block;">✓ Oficial</div> <p>Mostrar descripción...</p> <p>Desactivar <input type="checkbox"/> Activar solamente para grupos específicos</p>

Una vez habilitado el módulo nos vamos a la pestaña cifrado en administración y activamos el cifrado

Cifrado en el servidor *i*

☒ Habilitar cifrado en el servidor

Seleccione el módulo de cifrado predeterminado:

☒ Default encryption module

Modulo de cifrado por defecto

Tipo de cifrado: Cave de usuario

☒ Encriptar el almacenamiento personal

Al activar esta opción se encriptarán todos los archivos almacenados en la memoria principal, de lo contrario serán cifrados sólo los archivos de almacenamiento externo

Desactiva la clave de recuperación

La clave de recuperación es una clave de cifrado extra que se usa para cifrar ficheros. Permite la recuperación de los ficheros de un usuario si él o ella olvida su contraseña.

Contraseña de clave c

Repita la contraseña c

Desactiva la clave de recuperación

Importante: Los archivos que se encuentren en el servidor antes de la activación del cifrado permanecerán sin cifrar. Los archivos se cifrarán automáticamente tras subirse al servidor una vez esta opción sea activada.

Una vez hecho todo esto sería necesario montar un servidor de correo, ya que el administrador no podrá poner contraseña, se les enviaría un correo a los mismos de forma automática para que le asignen ellos mismos una contraseña.

6.1.4 Compartir una carpeta del servidor local

Esto es un paso opcional pero es una opción interesante que nos brinda ownCloud. Se trata de montar una carpeta del servidor para que su contenido sea accesible para el resto de los usuarios. Lo interesante es que solo será posible descargar ficheros de esa carpeta, no modificar ni subir ni borrar, solo el que tenga acceso físico al servidor puede modificar el contenido de esta carpeta. Esto puede ser útil como directorio de recursos públicos, aquellos ficheros que deben estar disponibles para todos pero que no pertenecen a nadie y nadie debe de poder modificar su contenido ni borrarlo.

Para habilitar esto primero debemos de editar el fichero de configuración principal de ownCloud, es decir, config.php (que se encuentra en el directorio de owncloud en la carpeta config) y añadir esta directiva

```
'files_external_allow_create_new_local' => true,
```

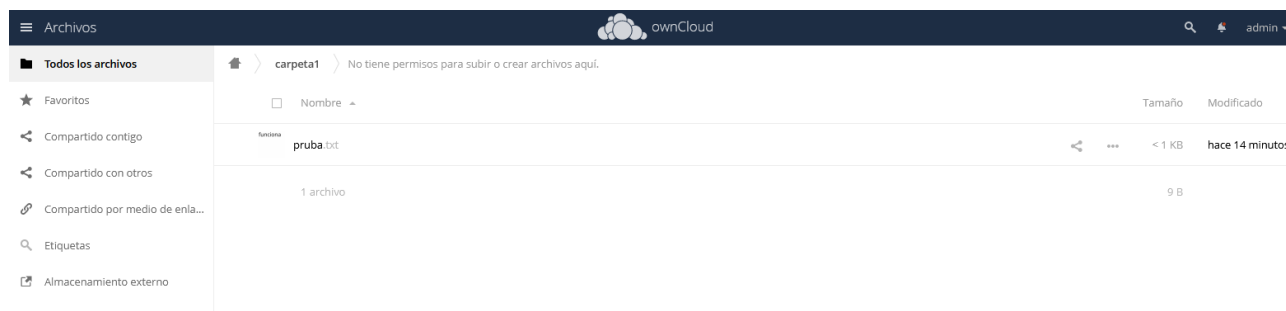
Reiniciamos apache y entramos en ownCloud como administrador. Dentro de ajustes en la pestaña de administración seleccionamos almacenamiento, marcamos “activar el almacenamiento externo” y recargamos la página. Una vez hecho esto añadimos como almacenamiento externo de tipo “local” una carpeta de nuestro sistema (debe de estar creada de antemano esa carpeta)

Almacenamiento externo

☒ Habilitar el almacenamiento externo

Nombre de la carpeta	Almacenamiento externo	Autenticación	Configuración	Disponible para
<input type="text" value="carpeta1"/>	Local	Ninguno ▾	<input type="text" value="/media/carpeta1"/>	<input type="text" value="admin x"/>
<input type="text" value="Nombre de la carpeta"/>	<input type="button" value="Añadir almacenam"/>			

Una vez hecho esto la carpeta y su contenido será visible en la pestaña “archivos” y “almacenamiento externo”



6.1.5 Alta disponibilidad

Justificación

Como cualquier servicio web tenemos múltiples opciones para incluir alta disponibilidad. Teniendo en cuenta el objetivo de este proyecto y los usuarios que posiblemente tenga pensamos que no es necesario un sistema de alta disponibilidad a gran escala pero si algo que solvete fallos esporádicos y no deje trabajo importante en espera.

Modelo

El modelo propuesto consiste en un servidor de respaldo que sustituye al principal en caso de fallo. El servidor de respaldo tendrá su base de datos y su servidor apache con sus datos sincronizados con el servidor principal de forma que este sirve también de copia de seguridad de respaldo en el caso de que se pierdan datos en el servidor principal.

Requisitos

Para implementar este modelo necesitamos una segunda máquina con el mismo entorno ya que se va a instalar owncloud en esta máquina también junto con todos sus requisitos y dependencias. Solo es necesario instalar las dependencias de ownCloud y mysql y apache2 ya que lo que vamos a hacer es sincronizar los directorios apache con rsync y que la base de datos del servidor de respaldo sea esclava del principal y replique.

Sincronización de servidores apache

Usando la herramienta rsync hacemos que el servidor de respaldo copie los datos del servidor principal periódicamente. En concreto copiara todo el directorio de apache (o el directorio de owncloud en apache) y el directorio data (por defecto dentro del directorio web pero puede estar en otro lado si lo indicamos en la instalación).

Sincronización de bases de datos

Para la replicación de la base de datos hemos editado el archivo de configuración de mysqld. Con descomentar las líneas de server id y log bin es suficiente, server Id debemos de asignarles números distintos a ambas máquinas. También hay que asegurar que bind address no sea la ip de

loopback si no la ip privada de la red interna.

```
ubuntu owncloud alta disp (b4 install) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.9.3 /etc/mysql/mysql.conf.d/mysqld.cnf

# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
# other settings you may need to change.
server-id      = 2
log_bin        = /var/log/mysql/mysql-bin.log
```

Antes de hacer la replicación es importante indicarle al esclavo que se salte la tabla oc_auth token ya que esa tabla puede generar conflictos que hagan que se pare la replicación (ver problemas encontrados). Podemos hacerlo en la CLI de mysql escribiendo `CHANGE REPLICATION FILTER REPLICATE_IGNORE_TABLE = (owncloud.oc_auth token);`

Soluciones para acceder a los servidores dependiendo de si el principal esta caído

Para que se acceda al servidor de respaldo cuando el principal está caído se puede configurar **con** keepalived, con scripts, con DNS o incluso podemos hacerlo manualmente cuando el servidor principal se caiga cambiando direcciones ip. Podemos también habilitar el acceso a ambos siempre asignando distintos nombres de dominio a ambos (owncloud.gcap.org y owncloud2.gcap.org por ejemplo).

Identificar si estamos en el servidor principal o el de respaldo

Debemos considerar que puesto que el servidor de respaldo es esencialmente una copia de seguridad funcionando como servidor web este servidor es inapropiado para trabajar en él ya que no se guardan los cambios en el principal, es decir, ambos no se sincronizan mutuamente y no deben ya que esto puede causar problemas si 2 personas editan el mismo archivo en los 2 servidores. Para añadir un poco de claridad a los usuarios hemos editado el index de ownCloud para que avise si se utiliza el servidor de respaldo

```
if (${_SERVER[SERVER_ADDR]} == '192.168.200.218'){
echo "<h1>ESTAS EN EL SERVIDOR DE RESPALDO, LOS ARCHIVOS QUE SUBAS O EDITES SE PERDERAN. GUARDA UNA$
echo "<br><br>";
echo "Estas en el servidor de respaldo guarda una copia local de los archivos que edites";
}
```

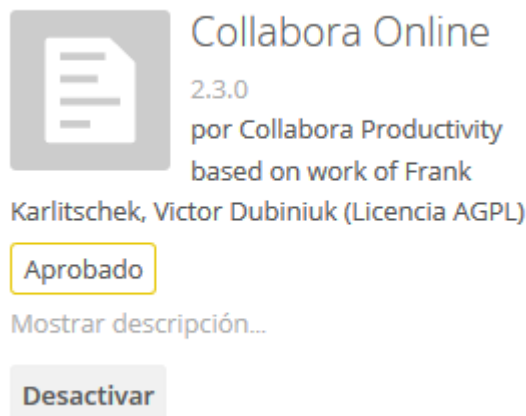
Lo cual mostrará el siguiente mensaje dentro de ownCloud



En este caso en concreto detecta que está en el servidor de respaldo por la ip pero también se puede hacer por nombre de host de la máquina.

6.1.6 Edición de documentos office con Collabora Online

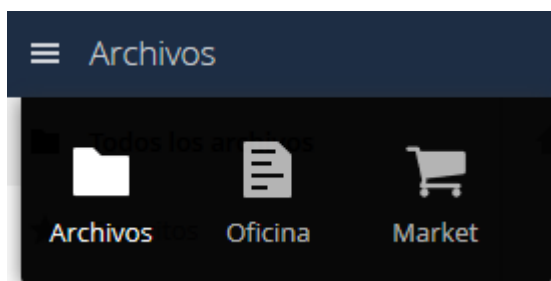
Para editar documentos office en línea necesitamos el plugin de Collabora para ownCloud y un servidor de Collabora Online. El plugin lo podemos obtener en el marketplace de ownCloud, se instala automáticamente. Una vez instalado dentro de ownCloud en el panel de administración dentro de aplicaciones lo podemos encontrar instalado.



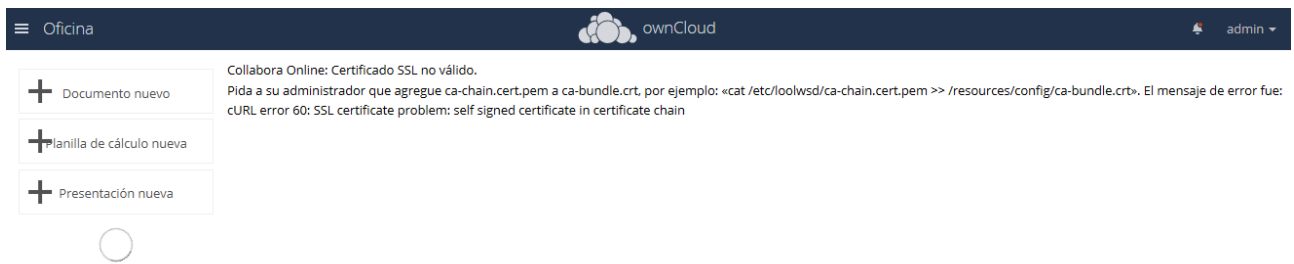
Una vez instalado el plugin instalamos Collabora, he decidido en el entorno de pruebas instalarlo mediante docker en la misma máquina de ownCloud. Con estos 3 comandos podemos tener Collabora Online funcionando.

```
-apt install docker.io (necesario ya que lo vamos a instalar con docker)
-docker pull collabora/code (nos descarga los binarios de Collabora para docker)
-docker run -t -d -p 127.0.0.1:9980:9980 -e "domain=localhost" -e "username=admin" -e "password=admin" --restart always collabora/code (Para iniciar el servicio de Collabora con Docker, en domain esta puesto localhost pero en un entorno real pondríamos nuestro dominio)
```

Ahora además de la pestaña archivos en ownCloud tenemos la pestaña oficina



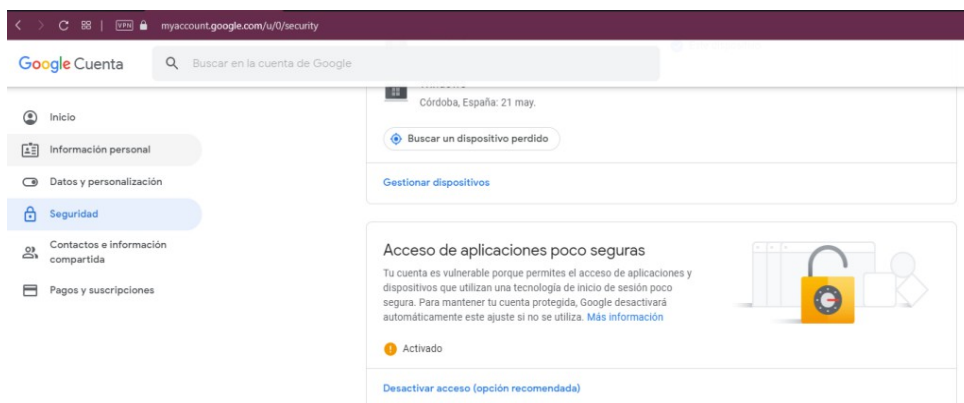
Es necesario tener certificados válidos para usar este servicio.



6.2 Configuración del servidor de correo

En el entorno de producción hemos configurado una cuenta de correo gmail para utilizar como servidor de correo. A través de la cuenta `owncloud@iesgrancapitan.org` se mandarían correos para administrar las contraseñas de las cuentas de usuario (tanto para asignar una contraseña tras crearse la cuenta como para restablecer la contraseña en caso de olvido)

La cuenta debe de ser configurada para que pueda ser utilizada por aplicaciones poco segura



Una vez hecho este cambio en la configuración de la cuenta debemos configurar el correo en ownCloud con los siguientes parámetros.

Servidor de correo electrónico *i*

Esto se usa para enviar notificaciones.

Modo de envío	smtp	Cifrado	SSL/TLS
Desde la dirección	owncloud	@	iesgrancapitan.org
Método de autenticación	Iniciar sesión	<input checked="" type="checkbox"/>	Se necesita autenticación
Dirección del servidor	smtp.gmail.com	:	465
Credenciales	owncloud@iesgranca	●●●●●●●●	Almacenar credenciales

6.3 Incorporar los directorios del NAS a la nube

Para compartir una carpeta de linux a linux la opción mas sencilla es hacerlo mediante NFS con el paquete `nsf-utils`. Añadiendo una entrada en el fichero `/etc/exports`. En este caso la carpeta ha sido compartida añadiendo esta línea a exports:

```
/share/MD0_DATA/homes/owncloud" 192.168.12.226(rw,async,no_subtree_check,insecure,no_root_squash)
```

Montamos la carpeta que ha sido compartida desde el NAS en nuestro sistema. Es necesario tener instalado el paquete nfs-common

```
root@owncloudserver:/home/owncloud# mount -t nfs 192.168.12.223:/share/MD0_DATA/homes/owncloud /localmount/
```

Debemos hacer que todo lo que cuelgue del directorio localmount sea propiedad de www-data para que ownCloud pueda gestionar los permisos de ese directorio.

En ownCloud en almacenamiento vamos a incluir el directorio montado que ahora esta en local como almacenamiento externo para la nube. Para que esto sea posible esta linea debe de estar en el fichero de configuración de ownCloud config.php







```
'files_external_allow_create_new_local' => true,
```

También se necesita tener instalado el paquete smbclient.

En ownCloud montamos la carpeta, queremos que el administrador tenga permisos de escritura y lectura y el resto solo de lectura. Por lo tanto la incluiremos 2 veces en ownCloud, una vez solo para el administrador con permisos de lectura y escritura y la segunda vez disponible para todos con permisos de lectura.

Almacenamiento externo

☒ Habilitar el almacenamiento externo

Nombre de la carpeta	Almacenamiento externo	Autenticación	Configuración	Disponible para	
 intranet	Local	Ninguno	/localmount	admin	 
 intranet_soloL	Local	Ninguno	/localmount	Todos los usuarios. Teclee para seleccionar	 
<input type="text" value="Nombre de la carpeta"/>	<input type="text" value="Añadir almacenam"/>				


☐ Permitir a los usuarios montar un almacenamiento externo



☒ Habilitar cifrado
☒ Establecer como solo lectura
☒ Habilitar previsualizaciones
☐ Habilitar el uso compartido
Comprobar si hay cambios Una vez cada acceso directo
☐ Compatibilidad con codificación Mac MFD (lento)

Si entramos en archivos y en la pestaña de almacenamiento externo (aunque en la principal también aparece) nos encontramos en el caso del administrador ambos accesos, tanto el de escritura como el de solo lectura (ya que el administrador es el único que puede escribir)

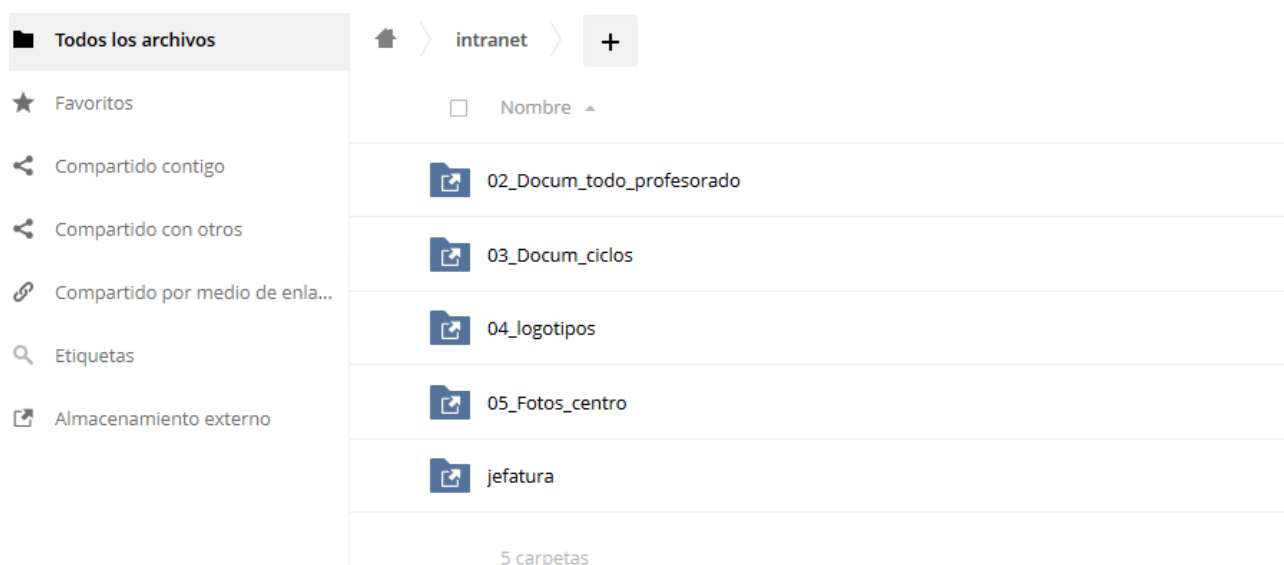
- ★ Favoritos
- Compartido contigo
- Compartido con otros
- Compartido por medio de enla...
- Etiquetas
- Almacenamiento externo**

Nombre ▲

 intranet

  intranet_soloL

2 carpetas



6.4 Blindaje y mejoras del servidor

Blindaje y mejoras del servidor

Una vez instalado ownCloud con HTTPS existen una serie de mejoras que se le pueden hacer al servidor para mejorar su seguridad y rendimiento. OwnCloud nos señala esto en configuración

Avisos de seguridad y configuración

- El bloqueo de archivo transaccional debería configurarse para utilizar el bloqueo basado en memoria, no el lento bloqueo predeterminado basado en base de datos. [Vea la documentación](#) para más información.
- Recomendamos activar el cron del sistema ya que otros métodos cron pueden influir en el desempeño y la fiabilidad.
- Su directorio de datos y sus archivos probablemente sean accesibles desde Internet. El archivo .htaccess no está funcionando. Le sugerimos encarecidamente que configure su servidor web de modo que el directorio de datos ya no sea accesible o que mueva el directorio de datos fuera de la raíz de documentos del servidor web.
- Este servidor no tiene una conexión a Internet. Esto significa que algunas de las características como el montaje de almacenamiento externo, las notificaciones sobre actualizaciones o instalación de aplicaciones de terceros no funcionan. Podría no funcionar el acceso a los archivos de forma remota y el envío de correos electrónicos de notificación. Sugerimos habilitar la conexión a Internet de este servidor, si quiere tener todas las funciones.
- La cabecera HTTP "Strict-Transport-Security" no está configurada en al menos "(segundos)" segundos. Para una mejor seguridad recomendamos que habilite HSTS como se describe en [security tips](#).
- La memoria caché no ha sido configurada. Para mejorar su desempeño, por favor, configure la memcache si está disponible. Puede encontrar más información en nuestra [documentation](#)

Por favor, compruebe de nuevo las [guías de instalación](#), y revise los errores o advertencias en el [archivo de registro](#)

HSTS

HSTS o HTTP Strict Transport Security nos permite obligar a que la comunicación sea solo por HTTPS con SSL/TLS lo cual nos puede ayudar a evitar ataques. Para habilitar HSTS debemos de incluir en nuestro virtual host de apache (el que lleva ssl) este apartado

```
<IfModule mod_headers.c>
Header always set Strict-Transport-Security "max-age=15552000; includeSubDomains"
</IfModule>
```

Protección del directorio data

Ademas de esta medida de seguridad también debemos de asegurarnos que no se puede acceder al directorio data de ownCloud. Si en la instalación de ownCloud elegimos un directorio seguro fuera de /var/www no seria necesario protegerlo ya que no es accesible desde fuera, sin embargo, si tenemos el directorio data dentro del directorio web o dentro de algún directorio accesible

desde fuera debemos restringir el acceso desde el exterior. Esto lo vamos a configurar con el archivo .htaccess de data. Primero configuramos apache para que use los archivos .htaccess. En el mismo virtualhost incluimos lo siguiente

```
<Directory /var/www/>
AllowOverride All
</Directory>
```

El archivo .htaccess de data debe de quedar así:

```
Generated by ownCloud on 2020-05-16 10:46:17
# line below if for Apache 2.4
<ifModule mod_authz_core.c>
Require all denied
</ifModule>

# line below if for Apache 2.2
<ifModule !mod_authz_core.c>
deny from all
Satisfy All
</ifModule>

# section for Apache 2.2 and 2.4
<ifModule mod_autoindex.c>
deny from all
IndexIgnore *
</ifModule>
```

Si alguien intenta acceder al directorio data apache debería de darle error 403 Forbidden o en su defecto mostrar una página en blanco vacía.

Bloqueo transaccional basado en memoria

Una vez mejorada la seguridad del sitio lo siguiente es el rendimiento. Lo mas notorio que podemos hacer para mejorar el rendimiento de la nube es cambiar el bloqueo transaccional basado en archivos por uno basado en memoria y habilitar una memoria cache para ello.

El bloqueo transaccional es un sistema que utiliza ownCloud para evitar que los archivos se corrompan cuando se suben o se modifican, bloquear todo el fichero supone una carga adicional sobre la base de datos. El bloqueo basado en memoria habilita una memoria cache que guarda los datos necesarios para evitar la corrupción de los ficheros. Para habilitar el bloqueo basado en memoria son necesarios los paquetes redis-server, php-redis y php-apcu. Una vez instalados debemos incluir en el fichero de configuración de ownCloud las siguientes líneas:

```
'filelocking.enabled' => true,
'memcache.local' => '\OC\Memcache\APCu',
'memcache.locking' => '\OC\Memcache\Redis',
'redis' => [
    'host' => 'localhost',
    'port' => 6379,
    'timeout' => 0.0,
    'password' => 'password', // Optional, if not defined no password will be used.
],
);
```

6.5 Alta disponibilidad con KeepAlived

Una vez implementada la alta disponibilidad igual que en el entorno de prueba hemos optado por redirigir las peticiones usando KeepAlived. KeepAlived es una herramienta que nos permite que 2

o mas máquinas compartan una IP virtual la cual apuntara a una máquina en concreto dependiendo de que servicios estén activos o caídos y la prioridad que le asignemos a cada una en la configuración.

Para ello necesitamos una IP libre en este caso va a ser la 192.168.12.228 que apuntara a un servidor u otro dependiendo de cual esté disponible, es decir nos redirigirá a la .226 y si esta no esta disponible a la .227

Instalamos el paquete keepalived en ambos servidores y creamos el archivo */etc/keepalived/keepalived.conf*

El archivo de configuración del servidor principal contendrá lo siguiente.

```
vrrp_script chk_apache2 {
script "pgrep apache2"
interval 2
}
vrrp_instance VI_1 {
interface ens160
state MASTER
advert_int 2
virtual_router_id 51
priority 100
authentication {
auth_type PASS
auth_pass miguelldavid
}
unicast_src_ip addr:192.168.12.226
unicast_peer {
192.168.12.227
}
track_script {
chk_apache2
}
virtual_ipaddress {
192.168.12.228
}
}
```

La segunda linea contiene la orden que servirá de condición para decidir si nos dirige a un servidor u a otro. En la mayoría de los ejemplos se usa pidof pero esto puede dar problemas en Ubuntu (ver la parte de KeepAlived en problemas encontrados)

En interface debe de estar el nombre de la interfaz de red con la que va a trabajar KeepAlived

State Master puede estar en ambos lo importante es la priority, el server que tenga mayor priority será al que nos dirija cuando ambos están activos. En el servidor de respaldo debemos de asignarle un valor menor.

unicast_src_ip addr es la IP de la máquina, en el servidor de respaldo pondremos ahí la ip del servidor de respaldo.

En unicast_peer en el principal esta la del servidor de resplado y en el de respaldo tendrá que estar la del principal

En virtual_ipaddress en ambos tiene que estar la IP virtual que responderá a las peticiones de ambos.

Adicionalmente ambos servidores deben de tener asignada esa IP en el caso de Ubuntu server hay que editar el netplan.

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    ens160:
      dhcp4: no
      addresses: [192.168.12.226/24, 192.168.12.228/24]
      gateway4: 192.168.12.1
      nameservers:
        addresses: [192.168.12.100, 8.8.8.8]
```

Basta con agregar la dirección aparte en addresses (no es necesario definir un nuevo interfaz de red)

Cuando paro el servicio apache2 se puede ver que en los logs del sistema keepalived marca el servidor como FAULT lo cual causa que redirija al servidor de respaldo si esta disponible.

```
Jun 11 18:46:00 owncloudserver systemd[1]: Stopped The Apache HTTP Server.
Jun 11 18:46:02 owncloudserver Keepalived_vrrp[6447]: /usr/bin/pgrep apache2 exited with status 1
Jun 11 18:46:02 owncloudserver Keepalived_vrrp[6447]: VRRP_Script(chk_apache2) failed
Jun 11 18:46:04 owncloudserver Keepalived_vrrp[6447]: VRRP_Instance(VI_1) Entering FAULT STATE
Jun 11 18:46:04 owncloudserver Keepalived_vrrp[6447]: VRRP_Instance(VI_1) Now in FAULT state
Jun 11 18:46:04 owncloudserver Keepalived_vrrp[6447]: /usr/bin/pgrep apache2 exited with status 1
```

6.6 Visor de PDF y Editor de .TXT

Instalar esta funcionalidad es muy sencillo, solo es necesario entrar en la pestaña de market (como administrador) y buscar el plugin de edición de texto y el de visor de PDF (se encuentran en la pestaña de productividad) con darle a instalar es suficiente ya que no requieren ninguna configuración.

7 Personal y Presupuesto

2 personas. Puesto que no es necesario pagar licencias de software ni adquirir nuevo hardware solo tendríamos que tener en cuenta la mano de obra de las 2 personas implicadas en el proyecto. Salen un total de 63,5h a razón de 25€/hora nos da un presupuesto de 1.587,50€

8 Grado de consecución de objetivos

N.º	Tareas	Cumplido (SI / NO)	Comentarios
01	Preparación Mvs para Entorno de pruebas	Si	En debian no funcionó
	Instalación y configuración de Owncloud	Si	

	Directorio de recursos públicos (carpeta compartida en servidor de solo lectura)	Si	
	Copias de seguridad	Si	
	Migración a Entorno producción	Si	
	Acceso a almacenamiento externo (Repositorio NAS)	Si	
	Configuración de una memoria cache y implementación del sistema transaccional basado en memoria	Si	
	Pruebas de verificación	Si	
02	HTTPS (Certificados Autofirmados)	Si	
	Edición de archivos : uso de Collabora Online y Letsencrypt	No	No funcionan en el entorno de producción (puertos)
	HTST	Si	
03	Configuración servidor de correo local (Roundcube)	Si	Sin TSL/SSL
	Cifrado	Si	
	Configuración servidor de correo en entorno de producción (gmail)	Si	
	Creación usuarios y grupos	Si	
	Implementar Oauth2 para autenticación usuarios de Google Gsuite	No	Parcialmente hecho
	Implementar autenticación contra un servidor LDAP	No	
04	Replicación en otra MV para alta disponibilidad.	Si	
	Instalación y configuración de KeepAlived	Aun no	
05	Protección del directorio data	Si	Mediante .htaccess
	Redirección automática de http a https	Si	

9 Competencias aplicadas

En este apartado se enumerarán las competencias del ciclo que se han aplicado a las distintas secciones de la integración del proyecto.

1. Servicios en la instalación básica de ownCloud (Su instalación es similar a algunos CMS como Wordpress)
2. Seguridad: Cifrado y SSL. Al incorporar HTTPS al servidor apache
3. Servicios: Apache. Aparte de SSL en apache tenemos hosts virtuales debido a que en las pruebas en local necesitamos varios servicios bajo una misma ip servidos por apache.
4. Servicios: Correo electrónico. En concreto hemos montado un servidor de correo con roundcube en local para que los usuarios asignen contraseñas y las recuperen.
5. Servicios y seguridad: Apache configuración de los .htaccess y de HSTS
6. Administración de sistemas operativos: NFS y NAS
7. Administración de sistemas operativos: Crontab, tareas automáticas con rsync para la alta disponibilidad
8. Base de datos: Modelo de replicación maestro-esclavo con mysql

10 Problemas encontrados

Versiones de PHP.

PHP 7.2 incompatible con owncloud 10.04, hay que añadir repositorios a la máquina para poder instalar php 7.0 junto a los módulos con la misma versión.

Trusted Domain.

OwnCloud tiene una directiva de seguridad en su fichero de configuración (por defecto está en su directorio de instalación en apache en la carpeta config) llamada trusted domains. Esta directiva indica como se puede acceder a ownCloud y se fija en la instalación. Por defecto se asigna ahí 1 única dirección ip o un único nombre de dominio y esa es la única forma de acceder a ownCloud. Aunque usemos otro nombre de dominio o ip que apunte al mismo servidor apache en la página de login de ownCloud obtendremos el error de dominio inseguro. Si cambiamos de ip o de nombre de dominio debemos cambiar esta directiva. Adicionalmente si tenemos distintos nombres de dominio o direcciones ip que apunten al servidor ownCloud los podemos incluir todos en la directiva ya que admite más de 1 nombre o dirección ip.

Edición de documentos office en línea

No se puede probar en local la edición de documentos office en línea. Se ha instalado un servidor de Collabora Online y su respectivo plugin en ownCloud, sin embargo, Collabora Online no acepta certificados apache autofirmados. Se puede usar la herramienta letsencrypt para obtener un certificado gratuito pero esta exige un dominio público.

KeepAlived: Configuración y uso de pidof

Por alguna razón desconocida si para comprobar que el servicio apache esta funcionando usamos la orden "pidof apache2" da un error en KeepAlived y no funcionará. Lo hemos resuelto usando "pgrep apache2" que da la misma salida pero por alguna razón no causa errores en KeepAlived

Alta disponibilidad: replicación, errores en las tablas

authtoken es una tabla de la base de datos de owncloud que guarda los tokens de sesión de los usuarios.

Si tenemos 2 bases de datos de owncloud una replicando de la otra podemos tener problemas si replicamos esa tabla. Si un usuario se loguea en ambos servidores causara un error de escritura ya que cuando se loguea en el servidor principal creara una entrada en la tabla la cual se copiara al servidor de respaldo, si el token no caduca y se loguea en el servidor de respaldo volverá a generar otro token para el mismo usuario sin que el otro haya expirado lo cual puede llegar a propagar errores en la base de datos hasta el punto de dejarla inservible.

Adicionalmente pueden surgir errores al hacer cambios en la configuración en el servidor de respaldo o subir archivos. En caso de que no se puedan solventar los errores y la replicación se pare es aconsejable borrar la base de datos del esclavo y sustituirla por una copia del maestro y reiniciar la replicación

Fallos al comprobar la integridad cache de los archivos


Este error puede deberse por varios motivos:

- Existen ficheros adicionales innecesarios dentro del directorio de ownCloud o del directorio data. Nos dirá que ficheros son, simplemente debemos borrarlos a mano.
- Faltan ficheros en el directorio de ownCloud. Owncloud nos dirá cuales faltan, es común que falten los ficheros .htaccess y .user.ini porque no se hayan copiado si hemos descomprimido ownCloud en otro directorio y luego lo hemos movido a www ya que estos archivos son archivos ocultos. Para arreglar este problema debemos copiar a mano estos 2 ficheros desde el tar.gz que nos descargamos de ownCloud en la instalación. No se pueden crear a mano.
- El reloj del sistema esta mal configurado.
- Los archivos están corruptos. Poco probable pero no imposible.
- Se han modificado archivos de forma fraudulenta (por ejemplo, si intentas cambiar un

archivo de ownCloud por otro “similar”).

Sincronización de carpetas de usuario mediante la aplicación de escritorio

Al igual que con Collabora no se puede utilizar esta funcionalidad por razones de seguridad. Debemos de utilizar certificados válidos

 Certificado sin verificar



No se puede conectar de forma segura a *cpd.iesgrancapitan.org*.

El certificado está autofirmado, y es no confiable

con certificado cpd.iesgrancapitan.org

Organización: Internet Widgits Pty Ltd

Unidad: <no especificado>

País: ES

Huella (MD5): 07:af:b4:44:03:78:b1:6f:54:8e:75:e1:35:a5:4c:2f

Huella dactilar (SHA1): 18:de:0f:6f:e3:20:19:48:72:77:24:fb:5c:36:f2:2b:f8:c5:96:63

Fecha de vigencia: ju. may. 21 15:24:35 2020 GMT

Fecha de caducidad: vi. may. 21 15:24:35 2021 GMT

Emisor: cpd.iesgrancapitan.org

Organización: Internet Widgits Pty Ltd

Unidad:

País: ES

Administradores de grupo: Pueden borrar a usuarios del sistema

Desconocemos si es una decisión de diseño intencional por parte del equipo de ownCloud o es un bug de la nube pero los administradores de grupo pueden borrar del sistema a los usuarios que tienen en su grupo, sin embargo, no pueden echarlos del grupo.

11 Futuras mejoras

OAuth2

OAuth2 es un protocolo de autenticación bastante usado por aplicaciones y proveedores. Permite autenticar a usuarios con credenciales de otras aplicaciones y servicios como google, microsoft, github etc... Para este proyecto era interesante loguear a los usuarios mediante google, ownCloud tiene un plugin para implementar OAuth2, sin embargo solo implementa la infraestructura de claves, se debe de configurar a mano las uri de redirección y el login.

LDAP

ownCloud soporta la autenticación de usuarios mediante LDAP con un plugin. Sería solo necesario tener un servidor LDAP funcional, instalar el plugin y configurar ownCloud para que se conecte al servidor LDAP.

Collabora Online

A falta de una forma de obtener certificados validos no hemos podido implementar el servidor de edición de documentos office en linea.

Personalización

Aunque es un poco rudimentario existen formas de personalizar algunos elementos de la interfaz de ownCloud, por ejemplo, la imagen de fondo que se muestra en la pantalla de login.

12 Bibliografía

https://doc.owncloud.org/server/10.4/admin_manual/

<https://tecadmin.net/setup-ip-failover-on-ubuntu-with-keepalived/>

<https://www.collaboraoffice.com/code/docker/>