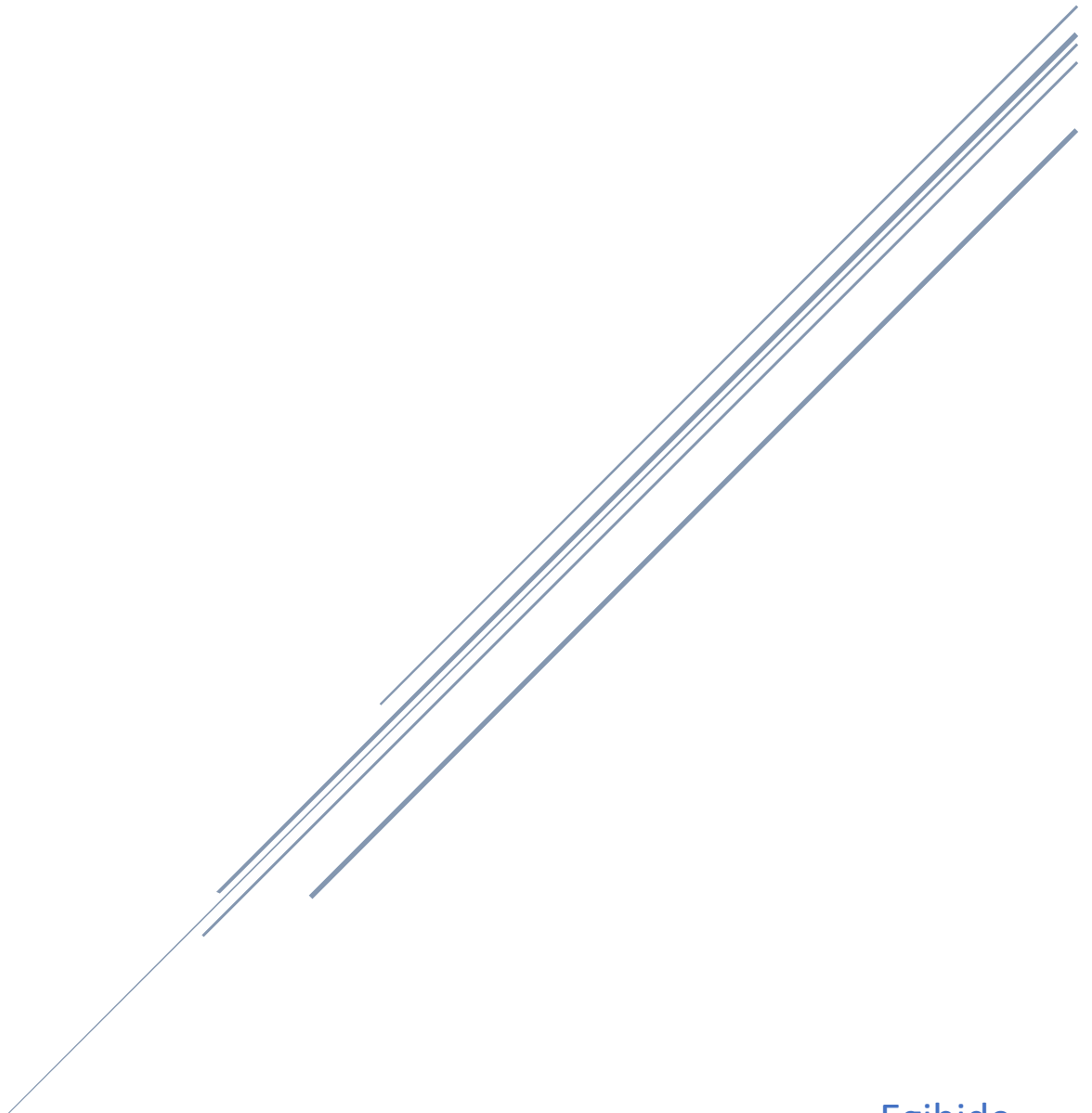


IMPLEMENTACIÓN DE WORDPRESS

Unai, Alejandro y Daniel (Grupo 2)

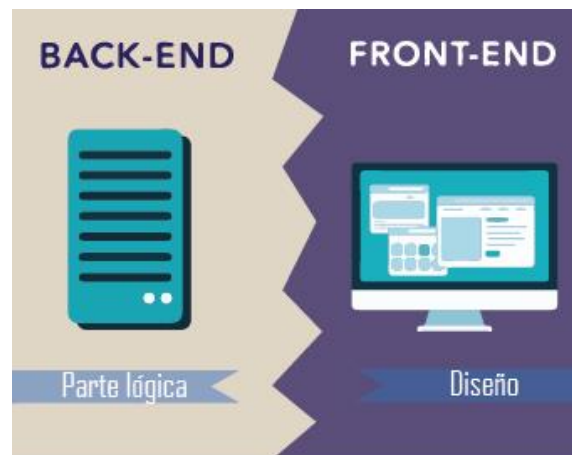


Egibide
1GDAW

Índice

Front end y back end	2
Estructura de directorios	2
Análisis de usuarios.....	3
Ficha técnica del servidor	3
Instalación	4
Configuración de WordPress	7
Análisis de la copia de seguridad	8
Licencia.....	8
Relación de etiquetas.....	8
Anexo: Creando las copias de seguridad del Blog	9

Front end y back end

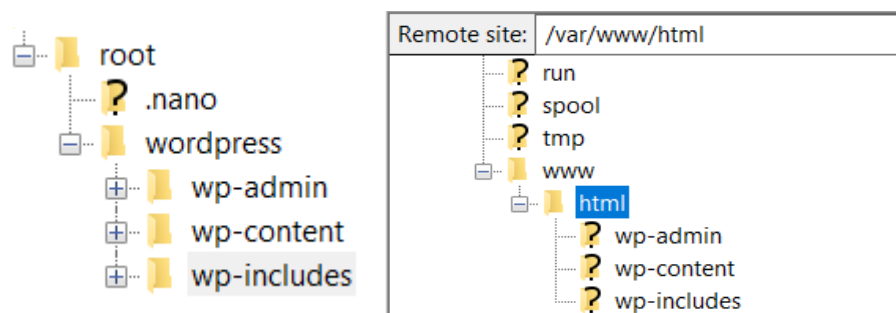


Front-end es un término que referencia a la capa de presentación. Esa capa es la que utiliza el usuario. Suele estar bien diseñada, cuidada y su objetivo es que sea fácil de usar para que el usuario no se pierda. Todas las interfaces que vemos a la hora de acceder a webs, navegar por el sistema operativo, programas, etc serían front-ends.

Back-end es un término que referencia a la capa de datos. Esta capa no es visible por el usuario y es donde ocurren las consultas, operaciones y demás acciones. Esta capa interactúa con el front-end.

Cuando usamos una web, algo tan básico como poner un usuario y contraseña, estamos utilizando un front-end (la página que vemos), ese front-end envía los datos que ha recogido la interfaz al back-end, el cual es el encargado de realizar las diferentes operaciones que harán que inicies sesión o no, es decir, el front-end es donde meterías tu usuario y clave, y, el back-end es lo que comprueba que ese usuario exista y que la contraseña sea correcta

Estructura de directorios



Al instalar WordPress se generan dos directorios importantes en `~/wordpress` y `/var/www/html`. En ambos se genera la misma estructura de directorios. “wp-admin” contiene todo el código referente al administrador de WordPress. “wp-includes” tiene en su interior todas las funciones de WordPress. “wp-content” alberga toda la gestión de un proyecto (temas, plugins, imágenes

subidas, ...) en WordPress se realiza en esta carpeta, exceptuando configuraciones generales, como el acceso a la base de datos.

Análisis de usuarios

Hemos concluido que vamos a necesitar dos usuarios del sistema operativo. El “root”, para configurar e instalar WordPress y otro usuario “userftp” para poder instalar plugins, temas y actualizar éstos mismos. Para conectarnos por SSH y poder trabajar más cómodamente utilizaremos el usuario root, dándole permisos:

```
Nano /etc/ssh/sshd_config
```

Quitamos el comentario de la siguiente línea (quitar el #)

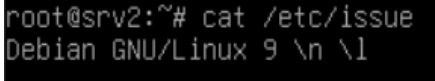
```
PermitRootLogin without-password
```

Y ya podríamos conectarnos introduciendo “ssh [root@10.1.2.1](#)” desde un terminal de Linux o “ssh 10.1.2.1” y posteriormente introduciendo el usuario “root” y su contraseña desde PowerShell.

Para la base de datos solo utilizaremos un usuario llamado “wordpress-user”, que será el que introduciremos en la instalación de WordPress a la hora de configurar MariaDB.

Para acceder a WordPress tenemos el usuario “admin” con el que gestionamos todo el sitio, y cada integrante del grupo tiene su propio usuario como autor, para poder editar entradas sin tener acceso a la configuración total del sitio, así evitar problemas.

Ficha técnica del servidor

Nombre	Srv2 (ID: 120)
Procesadores	1
RAM	1024
SO	Debian 9 
Usuarios	root (para usar el sistema y para ssh) instalacionWP (para instalar wordpress) user-ftp (para FTP)

	wordpress-user (para MariaDB) admin (para gestionar WordPress) Unaipuelles (autor) Alexddo (autor) Barraguesh (autor)
Networking	IP: 10.1.2.1/16 Gateway: 10.1.0.1
Instalaciones necesarias	PHP v7.0 (root) Net-tools v1.6 (root) Apache2 v2.4.25 (root)

Instalación

Comenzamos actualizando todos los componentes del sistema, tales como kernel, paquetes, parches de seguridad, ...

```
apt-get update
```

```
apt-get upgrade
```

```
apt-get dist-upgrade
```

Vamos a necesitar php así que instalamos todo lo necesario para poder ejecutarlo

```
apt install libapache-mod-php7.0 php7.0 php7.0-gd php7.0-xml php7.0-curl php7.0-mbstring php7.0-mcrypt php7.0-xmldrpc
```

```
apt-get install php7.0
```

Instalamos net-tools, que necesitaremos posteriormente para los comandos netstat

```
apt install net-tools
```

Instalamos MariaDB

```
apt install php7.0-mysql mariadb-server mariadb-client
```

Iniciamos MariaDB e instalamos

```
systemctl start mariadb
```

```
mysql_secure_installation
```

Instalamos apache2

```
apt install apache2
```

Comprobamos que apache2 ha sido instalado mirando que no haya mensajes de error durante la instalación y utilizando los siguientes comandos

```
systemctl status apache2
```

```
netstat | grep apache2
```

Activamos y configuramos el TLS para tener una conexión segura con el apache web server

```
a2enmod rewrite ssl
```

```
a2ensite default-ssl.conf
```

Editamos la configuración de apache en estos dos archivos

```
nano /etc/apache2/sites-enabled/000-default.conf
```

```
nano /etc/apache2/sites-enabled/default-ssl.conf
```

Incluimos este bloque de código en esos archivos, que permitirá a la aplicación de WordPress editar la configuración del propio apache. Algo necesario para la instalación.

```
<Directory /var/www/html>
```

```
Options Indexes FollowSymLinks MultiViews
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

Para poder implementar lo editado anteriormente activamos los “apache headers” y reiniciamos el servicio. Apache headers es un módulo que implementa el cambio de configuración introducido anteriormente.

```
a2enmod headers
```

```
systemctl restart apache2.service
```

Comprobamos si MariaDB ha abierto los puertos en “listening”, en caso de no tener el comando disponible.

```
netstat -tlnp | grep 'mysql'
```

Ya está LAMP configurado.

Acedemos a MariaDB como root.

```
mysql -u root -p
```

Creamos una database para instalar WordPress.

```
CREATE DATABASE wordpress;
```

Damos privilegios al usuario wordpress-user.

```
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress-user'@'localhost' IDENTIFIED BY 'wordpress-pass';
```

```
FLUSH PRIVILEGES;
```

Descargamos la última versión de WordPress, lo descomprimos y lo copiamos en el directorio `/var/www/html/`.

```
wget http://wordpress.org/latest.tar.gz
```

```
tar xzf latest.tar.gz
```

```
cp -rf wordpress/* /var/www/html/
```

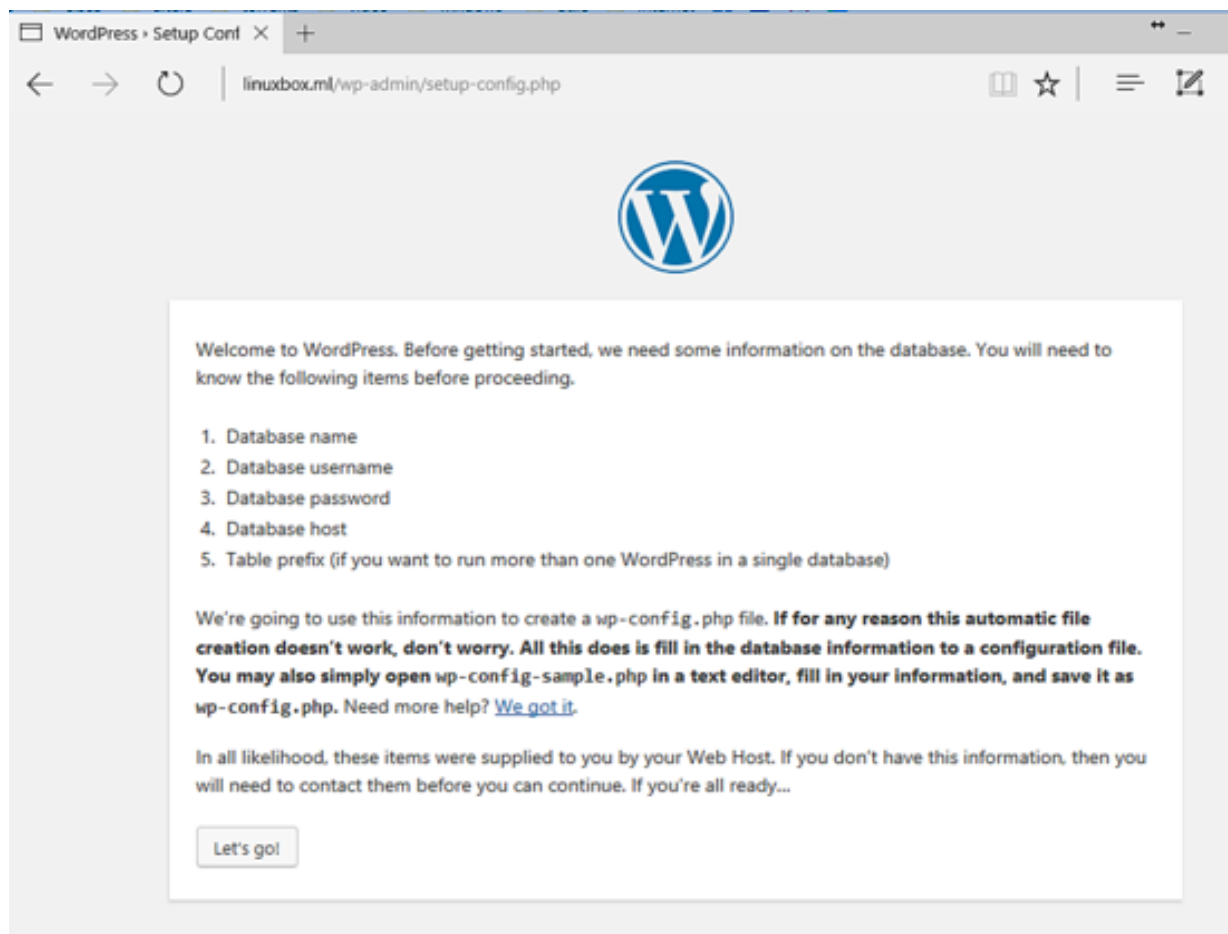
Damos permisos a las carpetas correspondientes para que WordPress pueda instalarse al iniciar la instalación en 5 minutos desde la web.

```
chmod -R 775 /var/www/html/
```

```
chgrp -R www-data /var/www/html/
```

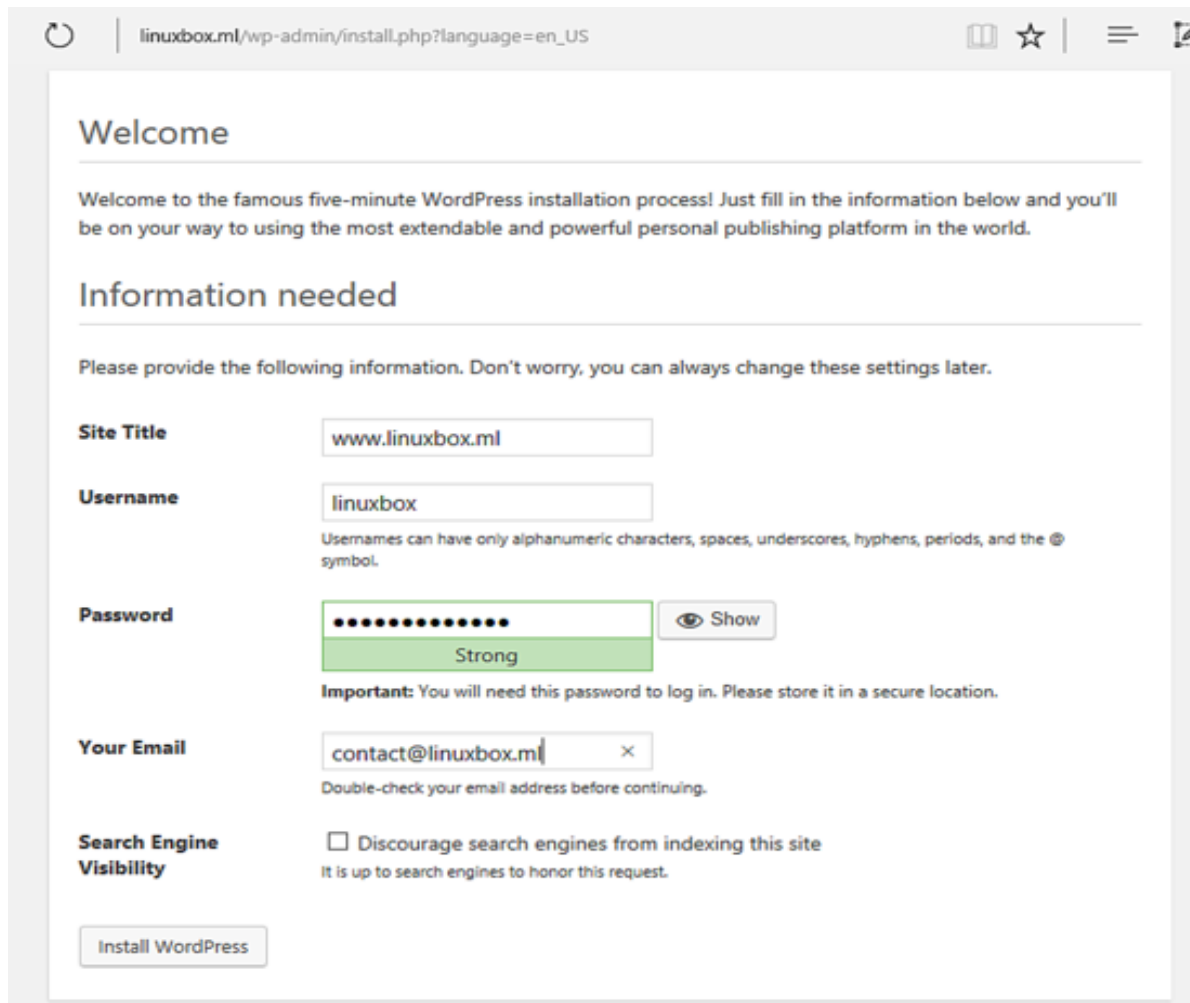
```
ls -al /var/www/html
```

Ya está listo lo necesario para usar WordPress y configurarlo desde página web: `10.1.2.1/wp-admin`. Desde donde accederemos a la instalación en 5 minutos de WordPress.



A: Configuración de la base de datos de WordPress

Y tras configurar la información de la base de datos de WordPress, configuramos el blog



B: Configuración en 5 minutos de WordPress

Configuración de WordPress

La configuración de WordPress es sencilla, es conocida por su “instalación en 5 minutos”. Nos pide conectarnos a la base de datos creada con MariaDB y posteriormente salen las opciones de configuración del sitio. Seleccionamos un nombre del blog, de usuario, una contraseña y si queremos que el sitio sea indexado por buscadores o no, continuamos y ya está configurado el blog. Ahora toca usar “wp-admin” para customizar el sitio, tocar ajustes o escribir entradas.

Para customizar y dar personalidad a nuestro blog, hemos decidido integrar plugins y un tema personalizado.

Hemos instalado dos plugins. Uno para hacer backups (explicado en el siguiente apartado) y otro para integrar configuraciones de idioma en el blog, es decir, crear dos entradas independientes en dos idiomas, y se mostrará solo la entrada en el idioma elegido (se aplica a todas las futuras entradas también).

Análisis de la copia de seguridad

Una copia de seguridad siempre es beneficiosa para evitar pérdidas inesperadas, pero es mejor tener varias copias con diferentes métodos para no depender solo de una. Hemos concluido que necesitamos hacer un backup de la máquina Debian y otra del propio WordPress, para tener dos pilares en los que sostenerse en caso de encontrar problemas. Haremos la copia de seguridad de WordPress usando el plugin UpdraftPlus. Hemos elegido este plugin porque era el más descargado (sacaba más de 1000 descargas a su competencia) y tenía las valoraciones más altas de otros usuarios.

El plan de actuación es el siguiente. En caso de fallo del servidor podemos restaurar la copia del contenedor de Proxmox, y por si algún casual estas copias están corruptas o no se restauran correctamente, existe la opción de restaurar nuestro blog tras volver a instalar WordPress en el servidor, usando la copia de seguridad del plugin. Además, en caso de tener problemas, las herramientas que hemos elegido disponen de notificación por email para poder solucionar el problema lo antes posible, o al menos ser conscientes de que necesitamos hacer una copia de seguridad manual porque la automática ha fallado.

Licencia



C: Licencia del Blog

Hemos decidido otorgar al blog una licencia creative commons con reconocimiento al autor (nosotros) y compartirigual (distribuir los derivados o copias bajo la misma licencia). El uso comercial está permitido.

Relación de etiquetas

Hemos decido empezar usando la etiqueta “primera semana” para el segundo post del blog. Iremos usando la misma etiqueta dependiendo de la semana para actualizaciones puntuales y otras etiquetas como avisos o actualizaciones cuando hayamos logrados avances importantes o cambios en la aplicación.

Para enfrentarnos a la creación de los posts en inglés y español, hemos implementado un selector de idioma que añade “es” o “en” a la URL del sitio. Por defecto el sitio se muestra detectando automáticamente el idioma del navegador, pero existe un menú en la cabecera que permite cambiar entre español o inglés y mostrará todas las entradas disponibles en el idioma seleccionado.

Anexo: Creando las copias de seguridad del Blog

Hemos utilizado la herramienta incluida en Proxmox para hacer una copia de seguridad automatizada del servidor. Se realiza todos los días laborables a las 14:30, y almacena hasta 7 copias. Es la única opción que existía para hacer un backup del contenedor.

Edit: Backup Job

Node:
-- All --
Send email to:
luisdaniel.barragues@ikasko
Storage:
local
Email notification:
Always
Day of week:
Monday, Tuesday, Wednesday
Compression:
LZO (fast)
Start Time:
14:30
Mode:
Snapshot
Selection mode:
Include selected VMs
Enable:
☒

<input type="checkbox"/>	ID ↑	Node	Status	Name	Type
<input type="checkbox"/>	100	proxmox223	stopped	srvdeb9p13	lxc
<input type="checkbox"/>	110	proxmox223	running	srv1	lxc
<input checked="" type="checkbox"/>	120	proxmox223	running	srv2	lxc
<input type="checkbox"/>	130	proxmox223	running	srv03	lxc
<input type="checkbox"/>	140	proxmox223	running	srv4	lxc
<input type="checkbox"/>	150	proxmox223	running	srv5	lxc

? Help
OK
Reset

D: Herramienta de backups de Proxmox

Además, hemos instalado un plugin (UpdraftPlus) que hace una copia de seguridad automática cada 4 horas en Google Drive, manteniendo 2 copias a la vez (las más actualizadas). Estas copias se guardan en la cuenta de Egibide de Daniel, no se puede configurar el Team Drive como destino, pero si se pueden utilizar otros servicios alternativos, en caso de necesitar guardarla en otro lado, tales como Dropbox, OneDrive, etc. Está configurado también el Email de Egibide de Daniel para recibir notificaciones en caso de problemas con las copias de seguridad automáticas.

The screenshot shows the 'Settings' tab of the UpdraftPlus plugin. At the top, there are tabs for 'Current Status', 'Existing Backups (2)', 'Settings' (selected), 'Advanced Tools', and 'Premium / Extensions'. Below the tabs, there are two sections for backup schedules: 'Files backup schedule' and 'Database backup schedule'. Both are set to 'Every 4 hours' and 'retain this many scheduled backups: 2'. A note below these sections explains that the time can be fixed or configured for more complex schedules, with a link to 'use UpdraftPlus Premium'. The main section is titled 'Choose your remote storage (tap on an icon to select or unselect):' and displays a grid of 15 storage options: UpdraftPlus Vault, FTP, S3-Compatible (Generic), Dropbox, Microsoft Azure, OpenStack (Swift), Amazon S3, SFTP / SCP, DreamObjects, Rackspace Cloud Files, Google Cloud, Email, Google Drive (highlighted in blue), Backblaze, and Microsoft OneDrive. WebDAV is also listed at the bottom. A link at the bottom states 'You can send a backup to more than one destination with an add-on.'

Current Status Existing Backups (2) **Settings** Advanced Tools Premium / Extensions

Files backup schedule: Every 4 hours and retain this many scheduled backups: 2

Database backup schedule: Every 4 hours and retain this many scheduled backups: 2

To fix the time at which a backup should take place, (e.g. if your server is busy at day and you want to run overnight), or to configure more complex schedules, [use UpdraftPlus Premium](#)

Choose your remote storage (tap on an icon to select or unselect):

- UpdraftPlus Vault
- FTP
- S3-Compatible (Generic)
- Dropbox
- Microsoft Azure
- OpenStack (Swift)
- Amazon S3
- SFTP / SCP
- DreamObjects
- Rackspace Cloud Files
- Google Cloud
- Email
- Google Drive
- Backblaze
- Microsoft OneDrive
- WebDAV

[You can send a backup to more than one destination with an add-on.](#)

Google Drive

E: Plugin de backups de WordPress

Al tener todo ya preparado y configurado hemos puesto ya en marcha el plan de actuación en caso de pérdida de datos del blog o del servidor comentado en “Análisis de la copia de seguridad”.