

# Instalación de LAMP en una máquina virtual

S. Alonso

## 1 Introducción y prerequisites

En este documento se detallan los pasos para tener un servidor LAMP instalado en una Máquina Virtual que puede ser usado en las prácticas de la asignatura Sistemas de Información Basados en Web.

### 1.1 Prerrequisitos

La máquina virtual se configurará usando **Vagrant** y **Virtual Box**. Consulte la documentación en línea de ambos para instalarlos correctamente en su sistema. Este tutorial está desarrollado asumiendo una máquina HOST con un sistema operativo Linux, aunque debe haber pocas diferencias en otros Sistemas Operativos distintos (una vez instalados los prerrequisitos).

## 2 Instalación de la máquina virtual

Vamos a utilizar la distribución Ubuntu 18.04 en nuestra máquina virtual. Para ello ya existe una “Vagrant box” preconfigurada (<https://app.vagrantup.com/bento/boxes/ubuntu-18.04>).

Creamos una carpeta para el proyecto:

```
1 > mkdir SIBW
2 > cd SIBW
```

Inicializamos la máquina virtual:

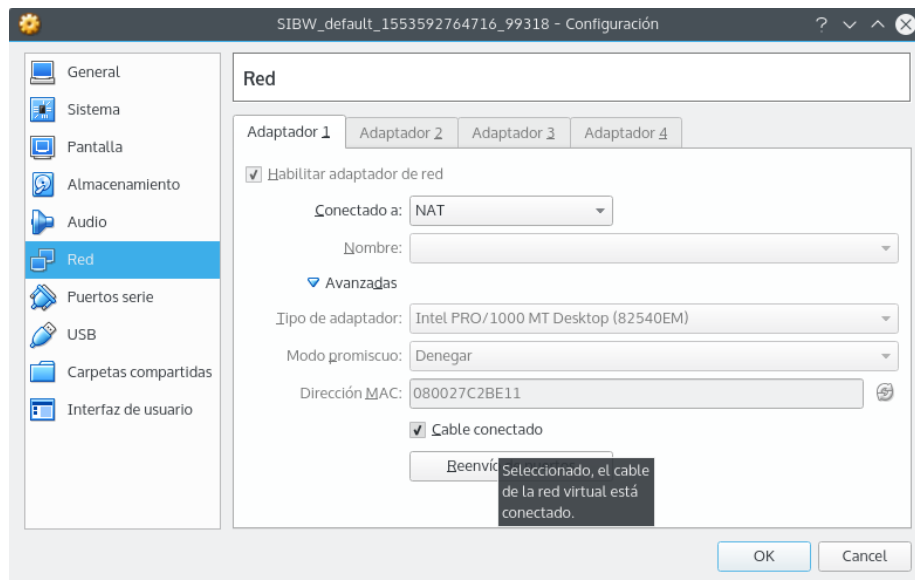
```
1 > vagrant init bento/ubuntu-18.04
```

Se creará un **Vagrantfile** en el directorio con la información necesaria sobre la máquina virtual.

Lanzamos la máquina virtual por primera vez (estando en el directorio del **Vagrantfile**):

```
1 > vagrant up
```

La primera vez que lancemos la máquina virtual tardará un rato porque tiene que descargar la imagen de Internet. En ocasiones es posible que este paso falle y la máquina no termine de arrancar o poder ser accedida. Hay que comprobar que la configuración de red de la M.V. es correcta. Para eso, en la Configuración de Virtual Box hay que asegurarse que está el “cable conectado” (**Configuración -> Red -> Adaptador 1 -> Avanzadas -> Cable Conectado**).



Una vez correctamente arrancada la máquina virtual podemos comprobar que tenemos un punto de montaje común: usualmente el directorio donde hemos creado el **Vagrantfile** estará montado en **vagrant** en la máquina virtual que acabamos de crear. Podemos comprobarlo:

```
1 > touch pruebaDirectorios.txt          # Creamos archivo
2 > vagrant ssh                          # Conectamos a la M.V. por SSH
3 MV> cd /vagrant
4 MV> ls
```

Se nos mostrarán los ficheros del directorio (como mínimo el **Vagrantfile** y **pruebaDirectorios.txt**). Recuerda que antes de apagar el ordenador es bueno apagar también la máquina virtual. Desde fuera de la máquina virtual podemos hacerlo con:

```
1 > vagrant halt
```

### 3 Instalación de la pila LAMP

Para tener configurada nuestra máquina virtual instalaremos **Apache**, **PHP** y **MySQL**:

```
1 MV> sudo apt-get update
2 MV> sudo apt-get install apache2
```

**Apache** por defecto escuchará en el puerto 80 de nuestra máquina virtual. Vamos a configurar la máquina virtual para que se haga una redirección del puerto 8080 de nuestro host al 80 de la máquina virtual.

```
1 MV> exit                                # Salimos de la MV
2 > vagrant halt
```

Editamos el **Vagrantfile** y añadimos:

```
1 config.vm.network "forwarded_port", guest: 80, host: 8080
```

Y volvemos a arrancar la MV:

```
1 > vagrant up      # En la salida comprobar que se hace la redireccion
2 > vagrant ssh      # Volvemos a entrar en la M.V.
```

El servidor **Apache** debe estar funcionando. Lo comprobamos accediendo en nuestro navegador a la dirección `http://localhost:8080`, es decir, nuestra máquina, puerto 8080, que está siendo redirigido al puerto 80 de la máquina virtual. Allí debemos ver la **Apache2 Ubuntu Default Page**. Seguimos instalando:

```
1 MV> sudo apt install mysql-server
2 MV> sudo mysql_secure_installation # Configuración inicial de MySQL
3                                     # Comprobar las opciones que
4                                     # aceptamos.
```

Con esto la BD debe estar configurada correctamente. Podemos entrar con nuestro usuario de **root** para crear usuarios con menos privilegios. **Importante:** NO debemos trabajar como **root** salvo para crear usuarios, bases de datos, etc. Seguimos instalando PHP:

```
1 MV> sudo apt install php libapache2-mod-php php-mysql
```

Para comprobar que efectivamente PHP está funcionando editamos el fichero de la máquina virtual: `/var/www/html/phpTest.php` y añadimos:

```
1 <?php phpinfo(); ?>
```

Y probamos a cargarlo en nuestro navegador: `http://localhost:8080/phpTest.php`. Debe aparecer la información de PHP. Continuamos configurando nuestro servidor para facilitarnos la vida: Vamos a hacer que **Apache** en vez de usar el directorio `/var/www/html` como base de donde servir nuestros ficheros use la carpeta compartida con el host: `/vagrant/html`:

```
1 MV> cd /vagrant
2 MV> mkdir html
```

Editamos el fichero `/etc/apache2/sites-available/000-default.conf` y cambiamos:

```
1 DocumentRoot /vagrant/html
```

Y editamos `/etc/apache2/apache2.conf` cambiando los permisos del directorio raíz:

```
1 <Directory /vagrant/html>
2     Options Indexes FollowSymLinks
3     AllowOverride None
4     Require all granted
5 </Directory>
```

Reiniciamos apache:

```
1 MV> sudo service apache2 restart
```

Creamos el fichero raíz de nuestro sitio web `/vagrant/html/index.php`:

```
1 <?php echo("HOLA") ?>
```

Y si todo ha ido bien al cargar en nuestro navegador: <http://localhost:8080> veremos:

```
1 HOLA
```

Con lo que habremos finalizado la configuración del servidor.

## 4 Uso del motor de plantillas Twig

Para instalar Twig haremos uso de **composer**, un gestor de dependencias de PHP.

Seguimos las instrucciones que nos encontramos en

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-composer-on-ubuntu-18-04>

```
1 MV> sudo apt install curl php-cli php-mbstring git unzip #Dependencias
2 MV> cd ~
3 MV> curl -sS https://getcomposer.org/installer -o composer-setup.php
4 MV> sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
5 MV> composer # Comprobamos que esta instalado
```

Instalamos Twig:

```
1 MV> cd /vagrant/html
2 MV> composer require "twig/twig:^2.0"
```

Y ya podemos crear un ejemplo básico.  
pruebaTwig.php:

```
1 <?php
2 require_once 'vendor/autoload.php';
3
4 $loader = new \Twig\Loader\FilesystemLoader('templates');
5 $twig = new \Twig\Environment($loader,[ ]);
6
7 echo $twig->render('pruebaTemplate.html',
8     ['nombre' => 'Espinete', 'edad' => 'Indefinida']);
9 ?>
```

templates/pruebaTemplate.html:

```
1 <!DOCTYPE html>
2 <html lang="es">
3     <head>
4         <meta charset="utf-8">
5         <title>title</title>
6     </head>
7     <body>
8         <p>{{nombre}} tiene edad {{edad}}.</p>
9     </body>
10 </html>
```

El directorio de nuestro proyecto tendrá un aspecto similar a:

