

## CONFIGURANDO LA COMPUTADORA RASPBERRY PI COMO SERVIDOR WEB

*M. en C. Cyntia E. Enríquez Ortiz*

*M. en C. Raúl Fernández Zavala*

*Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas (UPIITA)  
Instituto Politécnico Nacional*

### Resumen

*En la actualidad, para el desarrollo de aplicaciones web es necesario contar con un servidor HTTP (Protocolo de Transferencia de Hipertexto) instalado en alguna computadora, ya que este es el encargado de entregar la información solicitada por un cliente, el cual a su vez se encarga de mostrar esta información como una página web. Una buena alternativa para contar con un servidor web económico es utilizar la Raspberry Pi, la cual se puede configurar para que actúe como servidor HTTP de la aplicación web.*

### Servidores Web

Un servidor web o servidor HTTP es un programa que se ejecuta continuamente en una computadora, manteniéndose a la espera de peticiones de ejecución que le hará un cliente. El servidor se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página o información de acuerdo a los comandos solicitados.

Aunque la Raspberry Pi es menos poderosa que una computadora de escritorio o una laptop, no significa que no pueda utilizarse como un servidor web en un ambiente de negocios o educativo. A pesar de la pequeña cantidad de memoria y la relativa poca potencia del procesador, el bajo consumo de energía y el bajo costo de la Raspberry Pi hacen que sea una buena opción para entregar páginas estáticas o dinámicas a nivel de red local o incluso a través de Internet. Hay varias formas de lograr que la Raspberry Pi actúe como un servidor web, la más simple consiste en instalar alguno de los servidores web tradicionales, como Apache o lighttpd. Otra alternativa, es utilizar las herramientas que amplían los lenguajes de programación como Python, Ruby y JavaScript para crear servidores web que de forma dinámica generen el HTML cuando reciben peticiones HTTP desde un navegador web.

### Instalación de un servidor web Apache

La instalación del servidor web Apache es una buena opción si se quiere utilizar la Pila LAMP (combinación de Linux, Apache, MySQL y PHP). El uso de un servidor basado en LAMP, puede ejecutar aplicaciones que van desde los sistemas de gestión de contenidos a los foros interactivos, todo esto es posible realizarlo con la Raspberry Pi, siempre y cuando no se espere un rendimiento similar al de un servidor comercial de gran alcance.

Para poder instalar y configurar la pila LAMP se recomienda utilizar como sistema operativo Raspbian (distribución de Debian para la Raspberry Pi), posteriormente en la terminal o consola se utilizan los comandos `sudo apt-get update` y `sudo apt-get install apache2 php5 php5-mysql mysql-server` para instalar los paquetes necesarios. Una vez finalizada la instalación, tanto el servidor MySQL como el servidor web Apache se ejecutan en segundo plano (daemon). Para comprobar que el servidor funciona correctamente, se puede usar cualquier equipo que se encuentre conectado a la misma red donde está la Raspberry Pi y utilizando un navegador web, en la barra de direcciones se introduce la dirección IP de la Raspberry Pi, el navegador debe mostrar la página por defecto como se muestra en la figura 1.

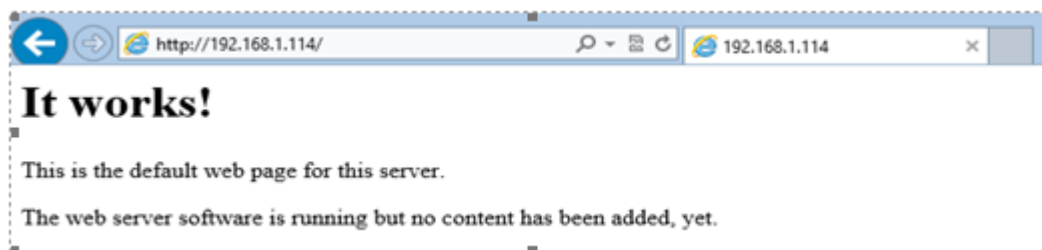


Figura 1. Página Web por defecto del servidor web Apache

El último paso es confirmar que el módulo PHP se instaló correctamente en Apache. Este módulo es importante porque es el que permite al servidor web ejecutar scripts PHP para entregar contenido dinámico, de otra forma el servidor web sólo podría entregar páginas estáticas. Para probar el funcionamiento correcto de este módulo, se puede crear un nuevo archivo script PHP, usando el siguiente comando escrito en una sola línea: `sudo sh -c 'echo "<php phpinfo ();?>"> /var /www /phptest.php'`. Este comando crea un nuevo archivo llamado phptest.php en el directorio /var/www, el cual le dice a PHP que cree una página de información para fines de diagnóstico. Para visualizar su contenido se utiliza un navegador en cualquier dispositivo conectado a la red y se introduce la URL <http://dir-IP-Raspberry/phptest.php>, también se puede probar en la misma Raspberry Pi escribiendo <http://localhost/phptest.php> en la barra de direcciones de su navegador.


<b>PHP Version 5.4.4-14</b> 	
System	Linux raspberrypi 3.6.11+ #371 PREEMPT Thu Feb 7 16:31:35 GMT 2013 armv6l
Build Date	Mar 7 2013 05:54:12
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysqli.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS
PHP Extension Build	API20100525,NTS
Debug Build	no
Thread Safety	disabled

Figura 2. Página Web dinámica entregada por el servidor web Apache

Uno de los inconvenientes que presenta el servidor web Apache en la Raspberry Pi es que su rendimiento no es muy bueno, es por eso que existen otras alternativas como el uso de servidores web ligeros.

### Instalación de un servidor web lighttpd10 (servidor web ligero)

Lighttpd10 es una buena opción para usarse en la Raspberry Pi debido a su bajo requerimiento de memoria. La instalación de este servidor es muy sencilla solo tiene que ejecutarse el comando `sudo apt-get install lighttpd`. Una vez completada la instalación el servidor está en funcionamiento y puede verificarse introduciendo desde cualquier navegador web la dirección IP de la Raspberry Pi como puede observarse en la figura 3.

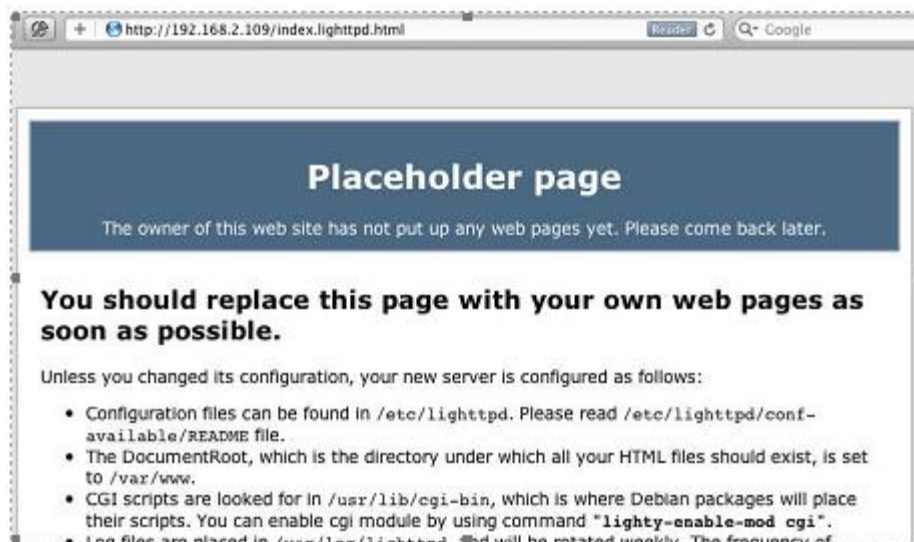


Figura 3. Página Web por defecto del servidor web Lighttpd

Con esta configuración el servidor es útil para entregar páginas estáticas, si se quiere entregar contenido dinámico, se necesita además algún lenguaje de programación, para la Raspberry Pi una buena elección es PHP, porque no usa muchos recursos, para instalarlo se utilizan los siguientes comandos:

```

sudo apt-get install update
sudo apt-get install php5-cgi
sudo lighty-enable-mod fastcgi
sudo /etc/init.d/lighttpd force-reload
    
```

Con esto se instala un intérprete de PHP y se activa el módulo FastCGI, en el servidor Lighttpd, el cual acelera los sitios web dinámicos. Para finalizar la instalación, se tiene que editar el archivo de configuración de Lighttpd (`sudo nano /etc/lighttpd/lighttpd.conf`) y añadir la siguiente línea para habilitar PHP y FastCGI.

```

fastcgi.server = (".php" => (("bin-path" => "/usr/bin/php-cgi","socket" => "/tmp/php.socket")))
    
```

Se deben guardar los cambios al archivo de configuración y reiniciar el servidor web, con el comando `sudo service lighttpd restart`. Para probar que todo trabaja correctamente, se puede crear un archivo como en el caso del servidor apache con el comando `sudo sh-c 'echo "<php phpinfo ();?>">/var/www/phptest.php'`, y utilizando cualquier navegador web mandarlo llamar para observar algo similar a la figura 4.

PHP Version 5.4.4-2	
System	Linux raspberrypi 3.1.9+ #171 PREEMPT Tue Jul 17 01:08:22 BST 2012 armv6l
Build Date	Jun 23 2012 12:17:43
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
Additional .ini files parsed	/etc/php5/cgi/conf.d/10-pdo.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525.NTS

Figura 4. Página Web dinámica entregada por el servidor web Lighttpd

## Instalación de un servidor web utilizando Flask

Otra alternativa para lograr que la Raspberry Pi actúe como un servidor web dinámico es usar el framework de Python llamado flask. Este framework, permite la autenticación de usuario, la generación de formas, y el uso de bases de datos, además de tener acceso a una gran variedad de bibliotecas estándar de Python que están disponibles.

Para instalar flask se debe instalar primeramente python para esto debe utilizar el comando *sudo apt-get install python-pip*, posteriormente se debe instalar flask con todas sus dependencias con el comando *sudo pip install flask*. Para probar la instalación, se puede crear un nuevo archivo llamado *hello-flask.py* con el siguiente código:

```

from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
    return "Hello, World!"
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80, debug=True)
    
```

En este código se crea un objeto *app* y se ejecuta el código posterior (en este caso solo se despliega "Hello, World!") cuando alguien acceda a la URL raíz del servidor. Después de esto se puede ejecutar el servidor, para lo cual se utiliza el siguiente comando *sudo python hello-flask.py*. Desde otro equipo que se encuentre en la misma red, se abre un navegador web y se escribe la dirección IP de la Raspberry Pi. Si el navegador muestra "Hello, World!", como se muestra en la figura 5 el servidor está bien configurado. Además en la Raspberry Pi se pueden observar las siguientes líneas:

```

10.0.1.100 - - [19/Nov/2012 00:31:31] "GET / HTTP/1.1" 200 -
10.0.1.100 - - [19/Nov/2012 00:31:31] "GET /favicon.ico HTTP/1.1" 404 -
    
```

La primera línea muestra que el navegador solicita la dirección URL raíz y que el servidor le regresa el código HTTP 200 para "OK". La segunda línea es una solicitud que muchos navegadores envían automáticamente para obtener un icono llamado *favicon*, para mostrarlo al lado de la URL en la barra de

direcciones del navegador. Como el servidor no tienen un archivo favicon.ico, devuelve el código HTTP 404 para indicar que no se encontró la dirección URL.



**Figura 5.** Página Web dinámica entregada por el servidor web Lighttpd

Una de las ventajas de flask es que se puede utilizar en conjunto con otras librerías de Python para proporcionarle funcionalidad adicional al sitio, por ejemplo con el módulo RPi.GPIO de Python se puede crear un sitio web que se comunique con el mundo real y controlar o monitorear cualquier variable dentro de una habitación.

## Conclusiones

La Raspberry Pi es una buena opción para actuar como servidor web en ambientes educativos, debido a su precio y flexibilidad, además ofrece varias alternativas para poder instalar y configurar un servidor web, dependiendo de las necesidades de la aplicación que se desea desarrollar, permite instalar servidores tradicionales como Apache que cuentan con acceso a una base de datos, servidores que tienen pocos requerimientos de memoria como Lighttpd el cual permite la entrega de contenido dinámico, o servidores que interactúan con el mundo real para controlar o monitorear variables mediante el uso de frameworks como flask.

## Bibliografía

1. Upton Eben, Halfacree Gareth, "Raspberry Pi User Guide", Wiley, 2012.
2. Upton Eben, Halfacree Gareth, "Meet the Raspberry Pi", Wiley, 2012.
3. Richardson Matt, Wallace Shawn, "Getting Started with Raspberry Pi", O'Really, 2012.