

Dpto. INFORMÁTICA – I.E.S. DOMINGO PÉREZ MINIK

MÓDULO PROYECTO

C.F.G.S. Administración de Sistemas Informáticos y en Red

Pi-Router

Autor: Javier Valencia Rodríguez

Fecha: 25 de Mayo de 2020

Tutor: José David Díaz Díaz

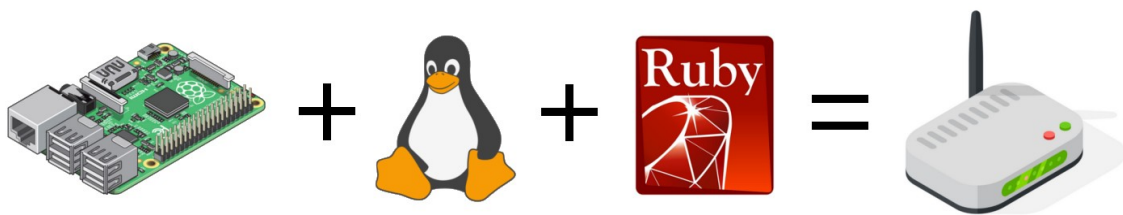
1. ÍNDICE

1. Índice.....	3
2. Introducción.....	5
3. Origen y contextualización del proyecto.....	6
4. Objetivos específicos.....	8
5. Análisis de un router de banda ancha.....	9
6. Implementación del hardware.....	12
7. Implementación del software.....	13
7.1. Sistema operativo.....	13
7.2. Aplicación web.....	15
7.3. Configuración.....	16
7.4. Microframework Sinatra.....	19
8. Implementación del router.....	21
8.1. Configuración inicial.....	21
8.2. Modo de trabajo.....	23
8.3. Firewall: reenvío de puertos y DMZ.....	24
8.4. Servidor de DHCP: dnsmasq.....	26
8.5. Punto de acceso WIFI: hostapd.....	27
8.6. Conexión a Internet: PPPoE.....	27
9. Instalación del software.....	30
9.1. Instalar Raspbian 10 en la Raspberry Pi.....	30
9.2. Instalación del software PiRouter.....	31

10. Interfaz web.....	33
10.1. Acceso inicial.....	33
10.2. Menú principal.....	34
10.3. Página de Sistema.....	34
10.4. Página de red WAN.....	35
10.5. Página de red LAN.....	37
10.6. Página de ajustes WIFI.....	38
10.7. Página de Rutas Estáticas.....	39
10.8. Página de Reenvío de Puertos.....	40
10.9. Página de Host DMZ.....	41
10.10. Página de Registros.....	42
11. Bibliografía.....	43

2. INTRODUCCIÓN

Este proyecto trata sobre el diseño e implementación de un router de banda ancha para el hogar, haciendo uso de un mini-pc Raspberry Pi, el sistema operativo GNU/Linux y el lenguaje de programación Ruby.



Este tipo de routers presentan una serie de características que los diferencian de los router tradicionales, y por este motivo resulta un tema interesante a desarrollar. En este documento se explicará, entre otras cosas:

- De dónde viene la inspiración para desarrollar este proyecto.
- Qué diferencias hay entre un router tradicional y un router de banda ancha.
- Qué beneficios e inconvenientes tiene el fabricar nuestro propio router.
- Qué debemos tener en cuenta al diseñar un router.
- Cómo configurar una Raspberry Pi.
- Cómo utilizar el microframework Sinatra para diseñar una aplicación web.

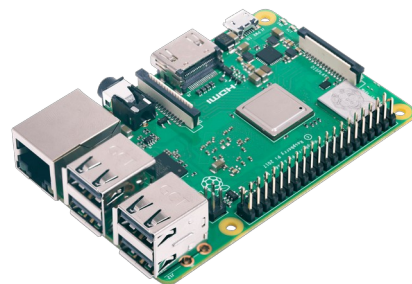
3. ORIGEN Y CONTEXTUALIZACIÓN DEL PROYECTO

Una de las materias que todo Administrador de Sistemas debe conocer es el enrutamiento de paquetes entre redes. Diversas asignaturas del ciclo tratan sobre ello y mi curiosidad me llevó a indagar un poco más en el tema.

De esta curiosidad nace este proyecto. ¿Qué mejor proyecto para entender el enrutamiento de paquetes que fabricar un router desde cero?. Hoy día es mas facil que nunca ya que los medios necesarios son muy pocos y, además, asequibles.

Desde el punto de vista del hardware se ha apostado por las Raspberry Pi, unos sencillos mini-pc más que competentes para la labor de un router. Estas máquinas cuentan con las siguientes características:

- Procesadores ARM de 32 y 64 bits.
- Memoria RAM desde 512MB hasta 4GB.
- Chip gráfico integrado.
- Tarjeta de red Ethernet integrada.
- Algunos modelos cuentan con chip WIFI integrado.
- Almacenamiento externo.
- Tarjeta de sonido integrado.
- Hub de puertos USB.
- Pines de expansión (GPIO).



Desde el punto de vista del software se va a utilizar el lenguaje de programación Ruby, ya que vengo usándolo desde hace algunos años y resulta muy sencillo de

entender. A su vez, se podría haber optado por un framework web como Rails, pero este tipo de frameworks es bastante “pesado” para la sencilla labor de un interfaz web de administración, por lo que me decanté por un microframework llamado Sinatra. La principal diferencia entre Rails y Sinatra es que este último es muy sencillo y ligero.

4. OBJETIVOS ESPECÍFICOS

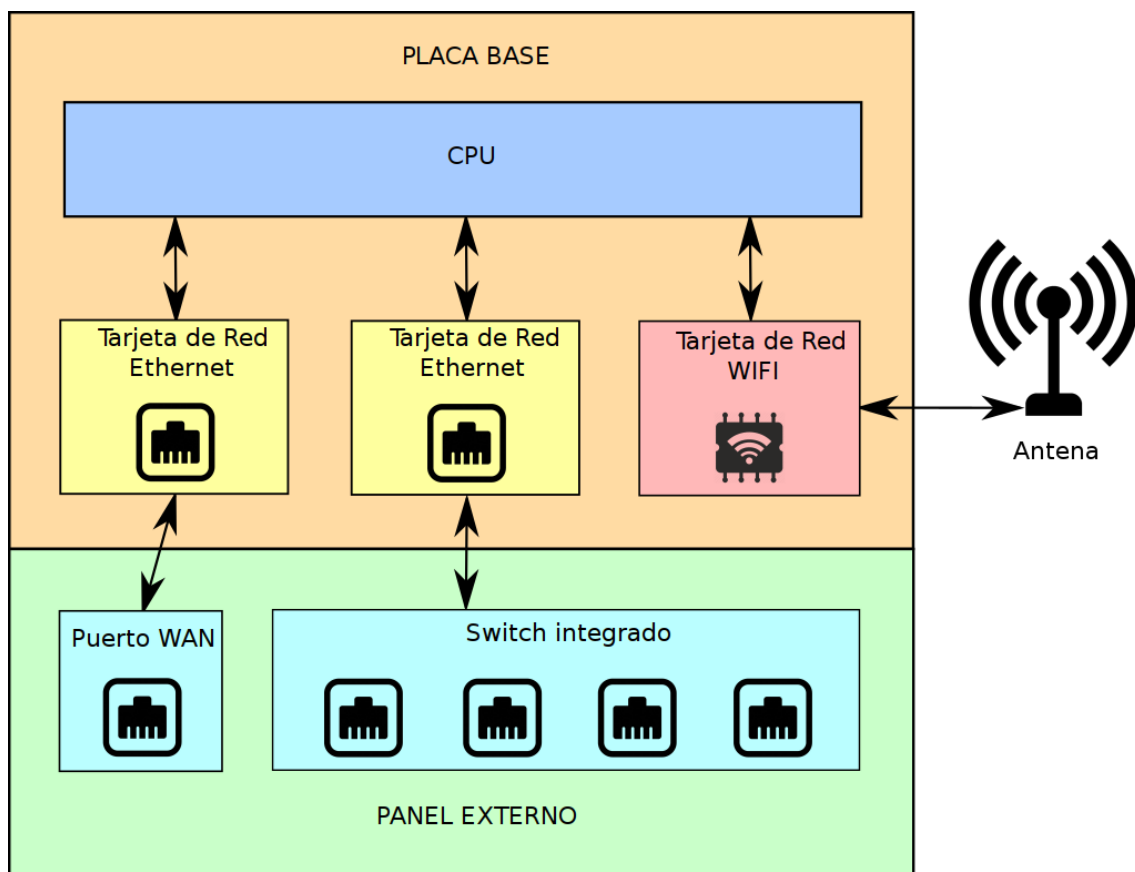
La siguiente lista resume los objetivos específicos del proyecto:

- Configurar el sistema operativo GNU/Linux de forma que actúe como un router.
- Mejorar mi conocimiento del stack de protocolos TCP/IP.
- Configurar distintos tipos de interfaces de red.
- Configurar un bridge.
- Configurar un firewall.
- Configurar un cliente de DNS.
- Configurar un servidor de DHCP.
- Configurar un punto de acceso WIFI.
- Configurar el acceso a Internet mediante el protocolo PPPoE.
- Configurar la Traducción de Direcciones de Red (NAT).
- Configurar el redireccionamiento de puertos a otros hosts.
- Configurar un equipo como host DMZ (zona desmilitarizada).
- Configurar rutas estáticas entre redes.
- Diseñar un interfaz web para la administración del router mediante el lenguaje de programación Ruby y el microframework Sinatra.
- Crear un script Bash de instalación del software.

5. ANÁLISIS DE UN ROUTER DE BANDA ANCHA

Para el desarrollo del proyecto es necesario hacer un estudio de los elementos que integra un router de banda ancha. Para ello he utilizado recursos propios (router de banda ancha de Movistar) y recursos externos (simuladores web de Linksys).

En primer lugar, se ha analizado el hardware, que en la mayoría de los casos resulta idéntico entre fabricantes. A continuación podemos ver en una ilustración el desglose lógico de un router de banda ancha:



Vemos que este router integra una tarjeta de red Wi-Fi para la creación de un punto de acceso. La antena puede ser interna o externa dependiendo del modelo. También se integra un switch conectado a una de las tarjetas de red Ethernet para dar

conexión a la red de área local. Este punto es el que más difiere de un router tradicional, donde cada puerto es una interfaz Ethernet individual, si bien existen modelos comerciales más caros que si presentan esta funcionalidad.

Por otro lado vemos el llamado “*Puerto WAN*”, que no es más que una interfaz Ethernet en la cual podemos utilizar diversos protocolos de conexión punto a punto para el establecimiento de una conexión de datos entre nuestro hogar y el proveedor de Internet. Los routers tradicionales no suelen presentar puertos WAN.

En cuanto al software, he examinado tanto mi router personal como otros modelos gracias a un simulador web de la empresa Linksys. Este simulador lo podemos encontrar en la siguiente URL: <https://ui.linksys.com/>.

En resumen, estas son las características de software comunes que presentan estos routers:

- Administración de una red de área local y de una conexión a Internet.
- Administración de un servidor de DHCP.
- Administración de un punto de acceso Wi-Fi.
- Administración de rutas estáticas.
- Administración de reenvío de puertos y red/host DMZ.
- Administración mediante interfaz web.

Según el modelo de router, se presentan algunas características adicionales que este proyecto no ha cubierto, como son:

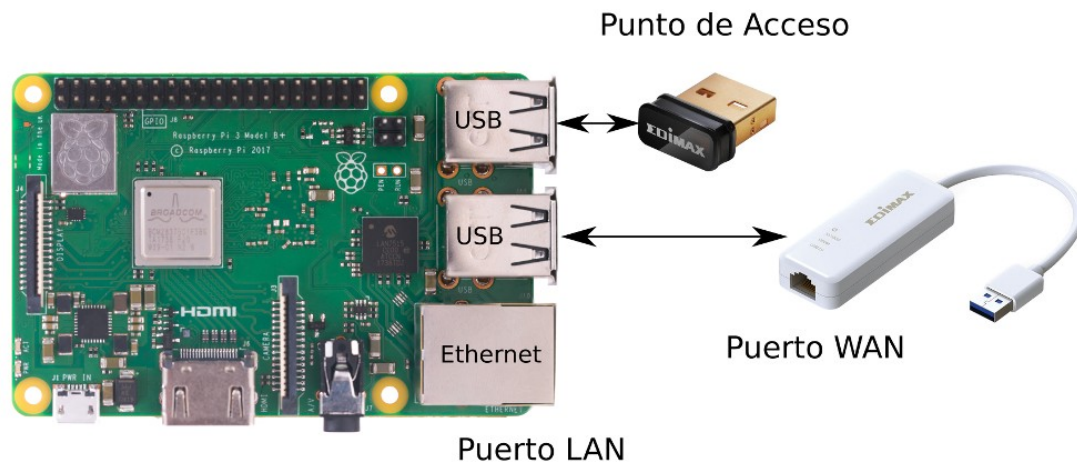
- Administración de VLAN en cada interfaz.
- Soporte para conexiones VPN.

- Protocolos punto-a-punto como L2TP y PPTP.
- Soporte para algoritmos de seguridad obsoletos como WEP y WPA-1.
- Filtrado MAC.
- Servidor de almacenamiento integrado.
- Restricciones horarias.

6. IMPLEMENTACIÓN DEL HARDWARE

A la hora de diseñar el router se ha optado por los siguientes elementos de hardware: una Raspberry Pi 3B, un adaptador de red Ethernet/USB y un adaptador de red Wi-Fi/USB.

El esquema resultante sería el siguiente:



El modelo de Raspberry Pi que se ha elegido es “3B”. Este modelo dispone de un chip Wi-Fi integrado, pero tiene poca potencia, si bien nos saca de un apuro; por eso se ha optado por el adaptador Wi-Fi/USB de Edimax, que tiene más potencia y rango.

El adaptador Ethernet/USB, también de Edimax, es un modelo con velocidades 10-100-1000 Mbits/s, aunque el puerto integrado de la Raspberry Pi 3B es de solo 100 Mbits/s. Eso significa que la velocidad máxima teórica del router será de 100 Mbits/s.

Por último, y sin mostrarse en la imagen, se necesita una tarjeta de memoria MicroSD y una fuente de alimentación de 5V y 2.4A.

7. IMPLEMENTACIÓN DEL SOFTWARE

7.1. SISTEMA OPERATIVO

El sistema operativo elegido es Raspbian 10 (GNU/Linux). Este es el sistema operativo recomendado por el fabricante de la Raspberry Pi y está basado en Debian 10. Existen otras alternativas como MS Windows 10 IoT, FreeBSD, etc. pero no disponen de tanto soporte ni tantos usuarios como Raspbian.

Dispondremos de los siguientes paquetes y herramientas para trabajar con el router:

Paquete	Herramientas
iproute2	ip, bridge
iptables	iptables
procps	sysctl, pkill
systemd	systemctl, hostnamectl, journalctl
ppp	pon, poff
dnsmasq	dnsmasq
hostapd	hostapd

El uso de cada herramienta es el siguiente:

- **ip**

Comando con el que configuramos las interfaces de red, asignando direcciones IP, de broadcast y máscara. También se utiliza para configurar un bridge entre varias interfaces. Es el comando más utilizado para administrar el router.

- **bridge**

Comando para modificar parámetros de un bridge.

- **iptables**

Comando para administrar el firewall de Linux. Nos permite filtrar paquetes, activar el encapsulado NAT, reenviar puertos y configurar zonas desmilitarizadas.

- **sysctl**

Comando para modificar parámetros del kernel. En nuestro caso se usará para activar el reenvío de paquetes en el kernel.

- **pkill**

Comando para matar procesos indicando su nombre. Lo he usado para administrar los programas “pon” y “poff” del paquete “ppp”.

- **systemctl**

Comando para la administración de servicios. Con el configuramos los servicios de DHCP (dnsmasq) y Punto de Acceso (hostapd).

- **hostnamectl**

Comando para configurar el nombre de equipo de la máquina.

- **journalctl**

Comando para examinar los registros de errores de Linux.

- **pon y poff**

Comandos que establecen o terminan una conexión PPPoE con el ISP.

- **dnsmasq**

Servicio de DNS y DHCP. En mi caso no uso el servidor de DNS.

- **hostapd**

Servicio que configura un Punto de Acceso WIFI.

7.2. APLICACIÓN WEB

La aplicación web es una aplicación Ruby modular diseñada con el microframework Sinatra, que hará uso de tres librerías diseñadas para trabajar con el sistema operativo del router. Para las vistas web se usa el lenguaje de plantillas “ERB” (HTML + Ruby), hojas de estilo CSS y lenguaje de scripting Javascript (jQuery). Para la instalación de la aplicación se ha creado un script en Bash.

La estructura de la aplicación es la siguiente:

```
config.yml
install.sh
pirouter
lib/
  config.rb  router.rb  service.rb
public/
  css/
    main.css
  js/
    jquery-3.4.1.min.js
views/
  dmz.erb  lan.erb  layout.erb  logs.erb  ports.erb  routes.erb
  system.erb  wan.erb  wifi.erb
```

La descripción de las librerías es la siguiente:

- **Router** (router.rb)

Implementa las funciones necesarias para configurar el router. Estas funciones modifican el sistema operativo según los deseos del usuario mediante la invocación de los comandos anteriormente descritos.

- **Config** (config.rb)

Implementa las funciones necesarias para trabajar con el fichero de configuración.

- **Service** (service.rb)

Implementa las funciones necesarias para trabajar con servicios Linux (arrancar, parar, reiniciar, habilitar y deshabilitar).

El microframework Sinatra requiere que la estructura de los directorios de la aplicación sea como la mostrada en el listado anterior (lib, views, public, ...). El código HTML de las paginas va en el interior de las plantillas (ficheros con extensión “.erb”).

La dinámica de la aplicación será la siguiente: cuando el usuario desee cambiar un parámetro del router a través de la interfaz web, primero se modifica el parámetro en el fichero de configuración (librería **Config**), y a continuación se modifica el estado del sistema operativo para reflejar los cambios (librería **Router**).

7.3. CONFIGURACIÓN

La configuración se lleva a cabo mediante la utilización de un fichero de tipo YAML. Se trata de un formato de datos legible por los humanos e inspirado por lenguajes como XML. La finalidad de este lenguaje es la de representar cualquier tipo de datos como un conjunto de listas (arrays), mapeos (hashes) y valores simples. Los documentos suelen empezar o dividirse mediante el uso de tres guiones (---) y es obligatorio el uso de la indentación mediante “espacios”.

La estructura del fichero de configuración (**config.yml**) se ha diseñado para separar los distintos parámetros del router en áreas o secciones temáticas. Dicha estructura se muestra a continuación:


```
---
devices:
  lan:
  wifi:
  wan:
router:
  mode: router
  hostname: pirouter
  login:
    username: admin
    password: "12345678"
  dns: []
lan:
  address: 192.168.1.1
  netmask: 255.255.255.0
  dhcp:
    enabled: false
    first_host:
    last_host:
    lease: 0
    dns: []
  ap:
    enabled: true
    ssid: pirouter
    password: "12345678"
    channel: 1
    hidden: false
    isolated: false
  wan:
    setup: none
    static:
      address:
      netmask:
      gateway:
    pppoe:
      username:
      password:
      vlan:
  routes: []
  port_forwarding: []
  dmz:
    host:
```

Ese sería el fichero de configuración con el mínimo de datos posible. Estos datos son: modo de funcionamiento del router, hostname, usuario y contraseña web, usuario y contraseña WIFI, dirección IP del router y su máscara. El resto de datos se modificarán desde la interfaz web.

El apartado “devices” debe rellenarse con los nombres de las interfaces de red antes de realizar la instalación. No se presenta rellenado porque depende de las interfaces de red añadidas a la Raspberry Pi. Este paso es recordado al usuario en el script de instalación.

La descripción de las distintas secciones es la siguiente:

- **devices**

Contiene la lista de dispositivos de red a usar por el router. Son tres:

- **lan**: la interfaz que actuará como red de área local para los clientes.
- **wan**: la interfaz con la que conectaremos a otras redes o Internet.
- **wifi**: interfaz wifi a usar como punto de acceso.

- **router**

Contiene parámetros básicos del router.

- **lan**

Contiene los parámetros de la red de área local, que incluyen también los parámetros del servidor de DHCP y del punto de acceso WIFI.

- **wan**

Contiene los parámetros de configuración de la interfaz WAN. Tiene un parámetro especial (**setup**) que determina si se configura estáticamente o por PPPoE.

- **routes**

Contiene las rutas estáticas añadidas al sistema.

- **port_forwarding**

Contiene los puertos redireccionados a otros equipos.

- **dmz**

Define el host DMZ de la red de área local.

7.4. MICROFRAMEWORK SINATRA

El microframework Sinatra resulta muy sencillo de usar y requiere muy pocos recursos. Por este motivo fue elegido para este proyecto. La forma de trabajar con este framework es definiendo una serie de funciones equivalentes a las rutas de nuestra página web, indicando el método de acceso a la URL y la propia URL (con o sin parámetros). Por ejemplo:

```
# Definición del índice principal "/" mediante el método GET
# Muestra "Hola mundo" en el navegador web cuando visitamos
# nuestra página "http://dominio.com/"

get "/" do
  return "Hola mundo"
end

# Método que recibe los parámetros de un formulario definido de
# la siguiente manera:
#   <form action="/comprobar_edad" method="POST">
#     <input type="text" name="edad">
#     <input type="submit">
#   </form>

post "/comprobar_edad" do
  if params['edad'].to_i > 30
    redirect "/mayor_de_30"
  else
    redirect "/menor_de_30"
  end
end
```

La utilización de plantillas también resulta muy sencilla. Se utiliza por defecto el lenguaje “ERB” que mezcla HTML y Ruby. Un ejemplo muy sencillo:

```
# Imprime una lista HTML numerada del nombre de los libros pasados
# en el array "books"

<ol>
  <% books.each do |book| %>
    <li><%= book.name %></li>
  <% end %>
</ol>
```

Podemos ver que se utilizan los tags “<% ... %>” para introducir código de control (bucles, if, etc.) y los tags “<%= variable %>” para obtener el valor de una variable. En caso de que esa plantilla se encuentre en el fichero “views/libros.erb” la forma de utilizarla en Sinatra es la siguiente:

```
# Suponemos la existencia de un modelo llamado "Libro" que
# retorna todos los libros de una base de datos.

get "/libros" do
  erb :libros, locals: {books => Libro.all}
end
```

El framework arranca su propio servidor web en el momento que ejecutamos la aplicación así que no es necesaria la instalación de uno.

8. IMPLEMENTACIÓN DEL ROUTER

8.1. CONFIGURACIÓN INICIAL

La librería “Router” implementa las funciones necesarias para configurar el sistema operativo como un router. Los pasos básicos para tener un router funcional son:

- **Inicializar las interfaces de red**

Las interfaces de red hay que habilitarlas al arrancar y eliminar cualquier dirección IP que tengan asignadas. Los comandos que nos permiten hacerlo son:

```
# Habilitar una interfaz de red
ip link set <interfaz> up

# Eliminar la configuración IP de una interfaz
ip address flush <interfaz>
```

- **Habilitar la redirección de paquetes en el kernel de Linux**

```
# Habilitar la redirección de paquetes
sysctl -w net.ipv4.ip_forward=1
```

- **Crear un “bridge” entre las interfaces LAN y Wi-Fi**

El router dispone de un punto de acceso Wi-Fi. Los clientes conectados a este punto de acceso deben encontrarse en la misma red que los clientes de la red de área local cableada. Así es como ocurre en la amplia mayoría de routers de banda ancha. En la aplicación se ha creado un dispositivo de tipo “bridge” que unifica a las interfaces LAN y Wi-Fi, y se le ha asignado la configuración IP que correspondería a la interfaz LAN. Los comandos para crear el bridge serían:

```
# Crear el bridge
ip link add <nombre-del-bridge> type bridge

# Habilitar el bridge
ip link set <nombre-del-bridge> up

# Asignar la interfaz LAN al bridge
ip link set <interfaz-lan> master <nombre-del-bridge>

# Asignar la interfaz WAN al bridge
ip link set <interfaz-wifi> master <nombre-del-bridge>
```

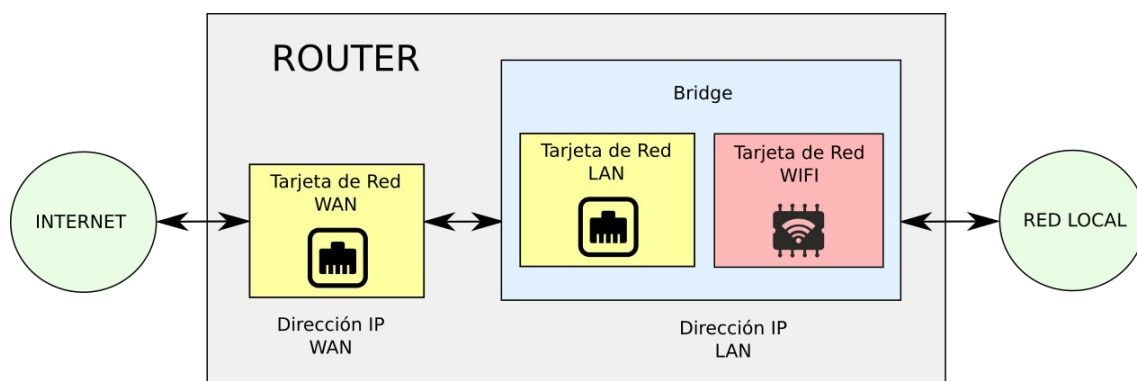
- **Asignar una dirección IP al bridge para activar la Red de Área Local**

```
# Asignar una dirección IP
ip address add <ip>/<máscara> broadcast + dev <bridge>
```

- **Asignar una dirección IP a la interfaz WAN**

La interfaz WAN se puede configurar de forma estática o mediante PPPoE. La finalidad última es que obtenga una dirección IP para poder comenzar a enrutar paquetes entre la red local y las redes externas. La asignación estática ya se ha comentado en el punto anterior y en el caso de usar PPPoE, es este protocolo el que negociará una dirección IP con nuestro proveedor de acceso.

Una vez que todas las interfaces se han configurado satisfactoriamente, el router está listo para enrutar paquetes entre las distintas redes. De forma opcional se pueden activar o desactivar el punto de acceso WIFI y/o el servidor de DHCP de la red local. El diagrama lógico del router (una vez configurado) es el siguiente:



8.2. MODO DE TRABAJO

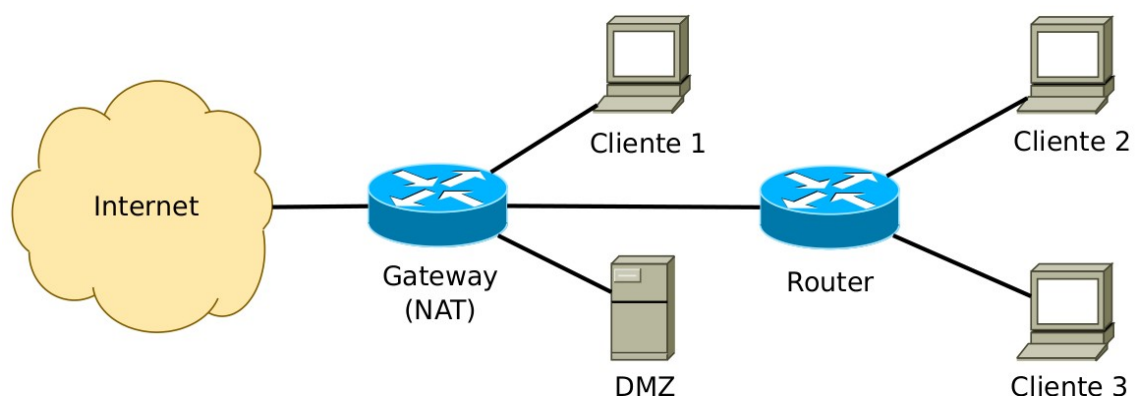
La mayoría de routers de banda ancha pueden desempeñar dos modos de trabajo: modo “router” y modo “gateway” (o pasarela).

Estos modos se diferencian según el punto de la red donde se encuentre el router y la función que vaya a cumplir. Un router tradicional se suele encontrar dentro de una red conocida donde podemos acceder a otros routers para establecer un enrutamiento de paquetes entre ellos. El tráfico de red es conocido.

Por contra, un router que actúa de gateway se encuentra siempre en los límites conocidos o administrativos de nuestra red, es decir, a partir de nuestro router no podemos administrar otros routers ni conocemos el tráfico que circula por allí.

Esto implica que un router en modo gateway necesitará forzosamente de un mecanismo para que los clientes de la red local puedan acceder a dichas redes desconocidas (siempre hablando en términos de IPv4). Este mecanismo es el sistema de Traducción de Direcciones de Red (NAT).

Un sencillo esquema muestra visualmente la diferencia entre ambos modos:



A parte del sistema NAT, un gateway también dispone de las siguientes funciones:

- **Reenvío de puertos**

Permite redireccionar puertos del gateway a otros hosts de la red local, con el fin de posibilitar la prestación de servicios por parte de un servidor, por ejemplo.

- **Host o Red DMZ**

Un host o red DMZ es aquel que recibe por defecto todo el tráfico externo que recibiría el gateway y que no se halla reenviado previamente a otro host mediante el reenvío de puertos.

- **Protección mediante firewall**

Al encontrarse en la frontera de nuestra red conocida, los gateway suelen disponer de un firewall para proteger la red.

Este proyecto implementa ambos modos de trabajo y todas las funciones de un gateway, dando la posibilidad al usuario de cambiar fácilmente entre un modo u otro.

8.3. FIREWALL: REENVÍO DE PUERTOS Y DMZ

El firewall de Linux es el encargado de implementar el sistema “*Source NAT*”, que nos permitirá navegar por redes externas (static NAT o Masquerade) y el sistema “*Destination NAT*”, que habilita el reenvío de puertos y el host DMZ.

Se denomina “*Source NAT*” a la traducción de direcciones de red que ocurre en el momento previo al envío de un paquete IP a una red externa. En este punto se modifica el campo “*Source IP*” del paquete asignándole la IP del gateway, de forma que los routers y hosts de Internet crean que el paquete viene del gateway (el cual dispone de una IP pública enrutable).

El firewall de Linux, **iptables**, implementa dos modos de “*Source NAT*”: el modo estático y el modo dinámico o “*Masquerade*”. En el modo estático, la IP que se asigna al campo “Source IP” siempre es la misma. Esto es recomendable cuando nuestro proveedor de acceso a Internet nos asigna una IP fija. En el modo dinámico o Masquerade, la IP que se asigna se obtiene del sistema operativo en el momento de asignarse. Esto es un proceso más lento que el modo estático pero es ideal para los casos en que no tenemos una IP fija asignada y por tanto puede cambiar en cualquier momento. En este proyecto sólo se ha implementado el modo dinámico, ya que funciona en ambos casos (IP fija y dinámica).

La forma de activar el modo estático sería la siguiente:

```
# Activar NAT dinámico en una interfaz
iptables -t nat -A POSTROUTING -o <interfaz> -j SNAT
--to-source <IP>
```

La forma de activar el modo dinámico sería la siguiente:

```
# Activar NAT dinámico en una interfaz
iptables -t nat -A POSTROUTING -o <interfaz> -j MASQUERADE
```

El sistema “*Destination NAT*” ocurre en el otro sentido, es decir, cuando un paquete IP va a ser enviado a un host de la red local. Llegado a este punto, se modifica el campo “*Destination Address*” del paquete para asignarle la dirección IP del host al que queremos reenviar el paquete. Este es el principio básico del reenvío de puertos y el host DMZ.

Para activar el reenvío de un puerto haríamos lo siguiente:

```
# Activar el reenvío del puerto 80 a un host de la red local
iptables -t nat -A PREROUTING -i <interfaz> -p tcp --dport 80
-j DNAT --to-destination <IP>
```

Para implementar un host DMZ utilizaríamos una regla como la anterior, pero sin indicar un puerto concreto (de forma que se aplique a todo el tráfico) y añadirla como la última regla de la cadena. Así podemos implementar el reenvío de puertos específicos a la vez que un host DMZ sin que hallan conflictos.

Para asignar un host de la red local como host DMZ haremos lo siguiente:

```
# Asignar un host como host DMZ
iptables -t nat -A PREROUTING -i <interfaz> -j DNAT
--to-destination <IP>
```

Por último, y como medida de seguridad básica, en el modo de trabajo “gateway” el firewall bloquea todo el tráfico entrante al router, es decir, aquel tráfico que no va a enrutarse, imposibilitando que un desconocido pueda acceder a los servicios propios del router (interfaz web, etc.).

Esto se lleva a cabo de la siguiente forma:

```
# Bloquear todo el tráfico entrante a una interfaz
iptables -A INPUT -i <interfaz> -j DROP
```

8.4. SERVIDOR DE DHCP: DNSMASQ

Para dotar a la red local de un servidor de DHCP se ha utilizado el servicio **dnsmasq**. Se trata de un servidor ligero de DHCP y DNS muy utilizado. La parte del servidor DNS se ha deshabilitado por no ser necesaria en un router. En la interfaz web se puede habilitar y deshabilitar el servicio así como configurar todos sus parámetros.

8.5. PUNTO DE ACCESO WIFI: HOSTAPD

El punto de acceso WIFI se ha creado con el servicio **hostapd**. Este servicio instruye a la interfaz de red indicada para que cambien al modo AP (access point). También controla todos los parámetros imaginables de un punto de acceso: autenticación, encriptado, ssid, canales, potencia, etc.

En este proyecto el punto de acceso sólo soporta el sistema de seguridad “WPA-2 Personal”, ya que “WEP” y “WPA-1” están obsoletos y “WPA-2 Enterprise” requiere de un servidor de autenticación RADIUS para funcionar, cosa que queda fuera del alcance de este proyecto.

8.6. CONEXIÓN A INTERNET: PPPOE

Tras el análisis de diversos routers de banda ancha, se llega a la conclusión de que los protocolos más usados para acceder a Internet con nuestros proveedores son: **PPP** y **L2TP**. Y de estos, al menos en España, PPP es el único utilizado en las conexiones del hogar. Por este motivo sólo se ha implementado el protocolo PPP en el software del router.

PPP (Point-to-Point Protocol) es un protocolo de comunicaciones de la capa de enlace (nivel 2 OSI) que se utiliza entre dos hosts de forma directa. Provee de autenticación, encriptado y compresión de datos. Se puede utilizar sobre diversos medios físicos, pero el medio físico que se utiliza por parte de los proveedores españoles es Ethernet. Esto le da el nombre de PPPoE (Point-to-Point Protocol over Ethernet).

La siguiente tabla muestra la localización del protocolo PPPoE en el stack de protocolos TCP/IP:

Aplicación	FTP	HTTP	DNS	SMTP	...
Transporte	TCP			UDP	
Internet	IPv4			IPv6	
Interfaz de Red	PPP				
	PPPoE				
	Ethernet				

El protocolo PPPoE añade un total de 8 bytes al contenido de las tramas Ethernet, 2 bytes pertenecientes a PPP y 6 bytes de PPPoE. Esto significa que cada paquete IP dentro de la trama Ethernet va a tener un tamaño de 1508 bytes, lo cual se aleja del estandar de 1500 bytes. Este tamaño máximo soportado se denomina “*MTU*” o Unidad de Transmisión Máxima, y en Ethernet se establece en los 1500 bytes antes mencionados. Ethernet también soporta otros tamaños de MTU en la transmisión, pero el problema es que no todos los equipos de red lo permiten por motivos de seguridad. Muchos routers en Internet rechazan paquetes que no sean de 1500 bytes ante la duda de tratarse de un paquete malformado por parte de un hacker.

Por consiguiente, cuando usamos PPPoE debemos restar 8 bytes de los 1500 bytes, es decir, 1492 bytes, de forma que al añadir los 8 bytes el paquete se adhiere al estandar y se puede enrutar sin mayor problema. Para arreglar este problema se aplican dos puntos: por un lado debemos especificar el tamaño de MTU que vamos a usar y por otro lado se utiliza el firewall de Linux para reajustar el campo “*MSS*” de los paquetes IP que pasan por el router. Este campo define el tamaño de los datos que viajan en el paquete IP y se calcula a partir de la MTU. El valor de MTU lo establecemos en el fichero de configuración de PPP.

Para habilitar este mecanismo debemos hacer lo siguiente:

```
# Habilitar TCP MSS Clamping
iptables -t mangle -A POSTROUTING -o <interfaz> -p tcp -m tcp
--tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Por último, cabe destacar que actualmente, gracias a la implantación de la fibra óptica, los proveedores de acceso están multiplexando sus canales de transmisión a través de VLAN. Esto significa que la transmisión de datos, televisión, voz y otros servicios se puede separar por VLAN (Movistar utiliza una VLAN para cada tipo de transmisión, mientras que otros proveedores usan una sola VLAN para todo el tráfico).

A la hora de conectarnos a Internet debemos configurar nuestra interfaz de red con la VLAN que el proveedor nos haya indicado. Sin ese dato no podremos utilizar dichos servicios.

Como muestra, aquí podemos ver algunas numeraciones VLAN utilizadas en España:

Proveedor	Datos (Internet)	Voz (VoIP)	Televisión (IPTV)
Jazztel	1074	--	838
Másmovil	20	--	--
Movistar	6	3	2
Orange	832	--	838
Vodafone	100	--	105

Para crear una interfaz virtual y asignarle un número de VLAN haremos lo siguiente:

```
# Habilitar una VLAN
ip link add link <interfaz-física> name <nombre-interfaz-vlan>
type vlan id <numero>
```

9. INSTALACIÓN DEL SOFTWARE

La instalación del software se divide en dos pasos: configurar una Raspberry Pi con Raspbian 10 y la instalación del software “*pirouter*”.

9.1. INSTALAR RASPBIAN 10 EN LA RASPBERRY PI

Para instalar Raspbian 10 en la Raspberry Pi vamos a seguir los pasos documentados en la página de [Instalación de Raspbian](#):

- Descargar la imagen
- Grabar la imagen en una tarjeta MicroSD
- Introducir la tarjeta en la Raspberry Pi
- Arrancar la Raspberry Pi y cambiar la contraseña por defecto

A continuación, si todo ha ido bien, accederemos a Raspbian bien por SSH o por modo físico mediante teclado y monitor.

El primer paso que debemos hacer es activar los “*nombres predictivos de interfaces de red*”. Este paso es muy importante ya que permite asignar un nombre fijo a las interfaces de red. Por defecto las interfaces de red se nombran de la siguiente forma: *eth0*, *eth1*, *eth2*, ..., y *wlan0*, *wlan1*, *wlan2*, ..., según se descubren las interfaces en el arranque. Esto significa que entre un arranque y otro, alguna interfaz de red puede cambiar de nombre y provocar problemas en el router.

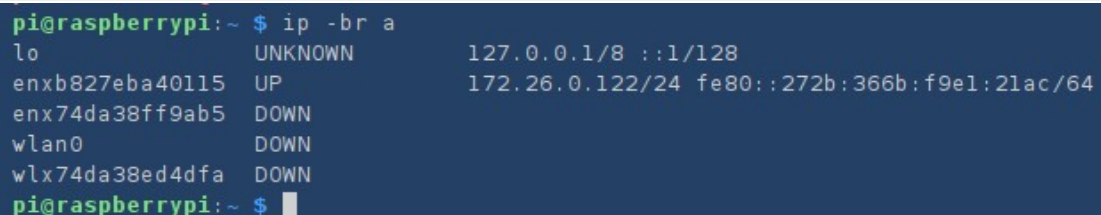
Activando los “*nombres predictivos de interfaces de red*” las interfaces dispondrán de un nombre único y constante. Este nombre estará formado por un sufijo que indica el tipo de interfaz de red (ethernet, wireless, etc.) seguido de su dirección MAC.

Por ejemplo, para el caso de las tarjetas de red ethernet tendremos el siguiente esquema: “**en****x**78e7d1ea46da”, donde podemos ver el sufijo “**EN**” de Ethernet seguido de una “**X**” que indica que a continuación viene la dirección mac “78e7d1ea46da”.

Para activar este sistema debemos eliminar el fichero situado en “/etc/systemd/network/99-default.link”.

```
# Activar nombres predictivos de interfaces de red
sudo rm /etc/systemd/network/99-default.link
```

Una vez hecho esto reiniciaremos y comprobamos que los nombres predictivos se encuentran activados:

A terminal window on a Raspberry Pi showing the output of the command 'ip -br a'. The output lists several network interfaces: 'lo' (UNKNOWN, 127.0.0.1/8), 'enxb827eba40115' (UP, 172.26.0.122/24), 'enx74da38ff9ab5' (DOWN), 'wlan0' (DOWN), and 'wlx74da38ed4dfa' (DOWN). The prompt is 'pi@raspberrypi:~ \$'.

9.2. INSTALACIÓN DEL SOFTWARE PIROUTER

La instalación del software es muy sencilla: debemos copiar el fichero comprimido con los datos de la aplicación a la Raspberry Pi mediante “scp” o usando un Pendrive USB. Una vez hecho esto, descomprimos el fichero:

```
# Descomprimir el software
tar -xzf pirouter.tar.gz
```

Ahora accedemos al directorio “*pirouter*” y ejecutamos el fichero de instalación: “*install.sh*” (se necesita conexión a Internet ya que se van a descargar paquetes):

```
pi@raspberrypi:~ $ ls
pirouter.tar.gz
pi@raspberrypi:~ $ tar -xzf pirouter.tar.gz
pi@raspberrypi:~ $ cd pirouter/
pi@raspberrypi:~/pirouter $ ls
config.yml  install.sh  lib  pirouter  public  views
pi@raspberrypi:~/pirouter $ sudo ./install.sh
#####
Programa de instalación de Pi-Router
(necesita conexión a internet)
#####

¿Ha configurado las interfaces de red en el fichero 'config.yml'?

s/n) █
```

Vemos que lo primero que nos indica es que debemos configurar las interfaces de red en el fichero de configuración “*config.yml*”. Para ello vamos a rellenar el apartado “*devices*” del fichero. Una vez rellenado, el apartado quedaría así:

```
GNU nano 3.2

---
devices:
  lan: enx827eba40115
  wifi: wlx74da38ed4dfa
  wan: enx74da38ff9ab5█
```

Los tres dispositivos DEBEN ser rellenados o el software no va a funcionar.

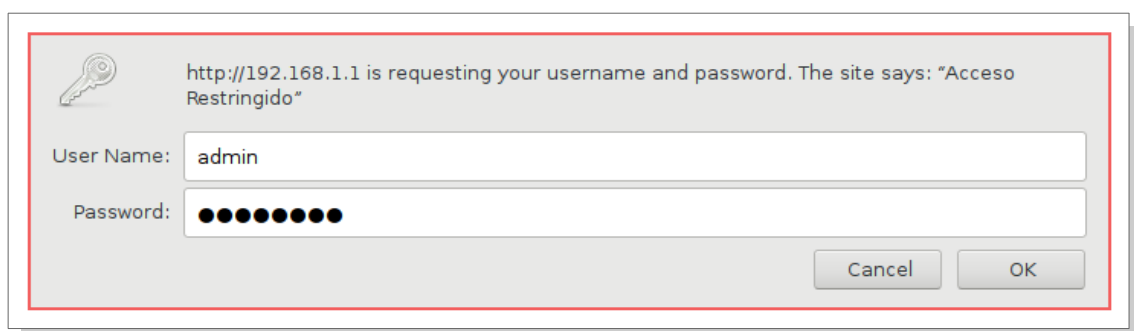
Volvemos a ejecutar el instalador y dejamos que termine de instalar. Una vez se ha instalado, el software residirá en el directorio “*/opt/pirouter*” y tan solo tendremos que reiniciar el sistema y acceder a la interfaz web.

10. INTERFAZ WEB

La interfaz web ha sido diseñada partiendo de un estilo simple y claro para el usuario. En este capítulo se va a describir cada una de las páginas que la componen.

10.1. ACCESO INICIAL

Una vez instalada la aplicación, el router toma la IP “192.168.1.1” por defecto, como viene siendo habitual en la mayoría de routers comerciales. Para acceder a la interfaz web tenemos que visitar la siguiente página en nuestros navegadores: <http://192.168.1.1>. Hay que recordar que debemos tener configurada una IP en nuestro PC/portatil/movil en la misma red que el router, para poder acceder a la interfaz web. Nada más acceder a la URL anterior, la interfaz web nos presenta el diálogo de acceso preguntandonos el nombre de usuario y la contraseña. Por defecto estos datos son: usuario “admin” y contraseña “12345678”.



10.2. MENÚ PRINCIPAL



La interfaz web presenta un menú principal desde el que podemos acceder a cualquiera de las páginas que necesitemos.

10.3. PÁGINA DE SISTEMA

The image shows the "SISTEMA" page of the Pi-Router web interface. The page has a green header with "Pi-Router" and a navigation bar with buttons for "SISTEMA", "WAN", "LAN", "WI-FI", "RUTAS", "PUERTOS", "DMZ", and "LOGS". The main content area is divided into several sections:

- Modo de trabajo**: A dropdown menu is set to "Router". To the right, text states: "En el modo **Gateway** se activan el sistema NAT, la redirección de puertos y el host DMZ." There is an "Aplicar" button.
- Datos opcionales**: A text input field for "Hostname *" contains the value "pirouter". There is a "Guardar" button.
- Servidores DNS**: Two text input fields for "Primario:" and "Secundario:" are both marked as "(opcional)". There is a "Guardar" button.
- Administración**: Four text input fields for "Contraseña actual *", "Usuario:" (containing "admin"), "Contraseña nueva:", and "Repetir contraseña:". There is a "Guardar" button.
- Sistema**: A section at the bottom containing three buttons: "Descargar Configuración", "Reiniciar", and "Apagar".

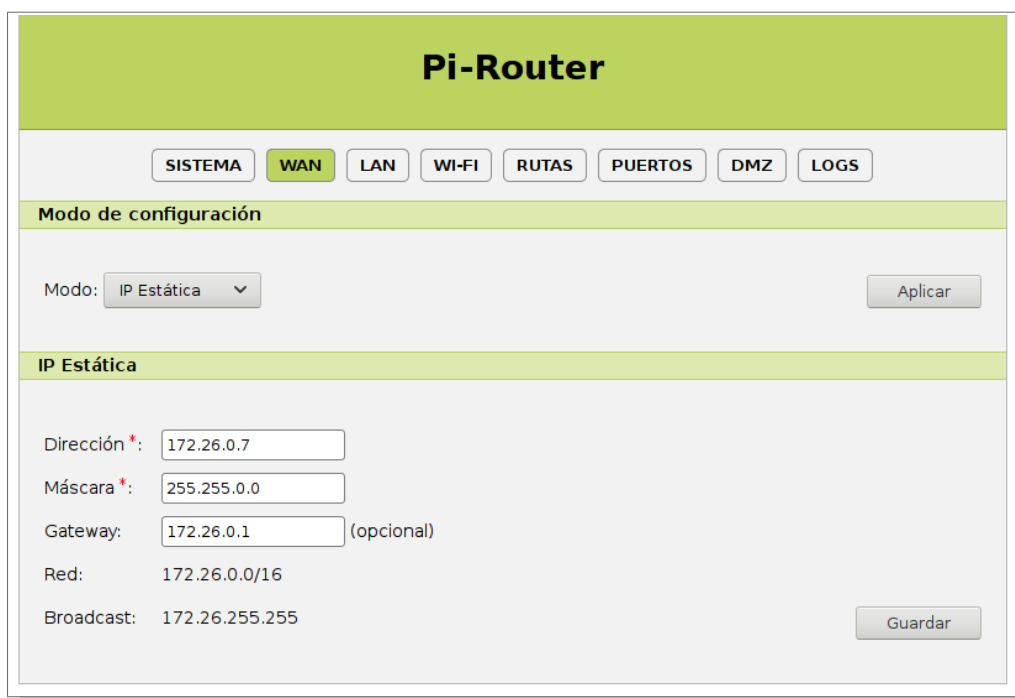
En esta página podemos cambiar el modo de trabajo del router, modificar el nombre de equipo, configurar los servidores DNS del router (para uso interno), cambiar el nombre de usuario y contraseña web, descargar el fichero de configuración para hacer una copia de seguridad, reiniciar y apagar el sistema.

10.4. PÁGINA DE RED WAN



The screenshot shows the Pi-Router web interface. At the top is a green header with the text "Pi-Router". Below it is a navigation bar with buttons: SISTEMA, WAN (highlighted in green), LAN, WI-FI, RUTAS, PUERTOS, DMZ, and LOGS. Under the navigation bar is a section titled "Modo de configuración". In this section, there is a label "Modo:" followed by a dropdown menu. The dropdown menu is open, showing three options: "Deshabilitada" (which is highlighted in blue), "IP Estática", and "PPPoE". To the right of the dropdown menu is a button labeled "Aplicar".

En modo “*Deshabilitada*”



The screenshot shows the Pi-Router web interface in the "IP Estática" mode. The header and navigation bar are the same as in the previous screenshot. The "Modo de configuración" section now shows the "Modo:" dropdown menu set to "IP Estática". Below this section is a new section titled "IP Estática". This section contains several input fields: "Dirección *:" with the value "172.26.0.7", "Máscara *:" with the value "255.255.0.0", "Gateway:" with the value "172.26.0.1" and the text "(opcional)" next to it, "Red:" with the value "172.26.0.0/16", and "Broadcast:" with the value "172.26.255.255". At the bottom right of this section is a button labeled "Guardar".

En modo “*IP Estática*”

The screenshot shows the Pi-Router web interface. At the top is a green header with the text "Pi-Router". Below it is a navigation bar with tabs: SISTEMA, WAN (highlighted), LAN, WI-FI, RUTAS, PUERTOS, DMZ, and LOGS. The main content area is titled "Modo de configuración". Under this, there is a "Modo:" dropdown menu set to "PPPoE" and an "Aplicar" button. Below this is a section titled "PPPoE". It contains three input fields: "Usuario:", "Contraseña:", and "VLAN:". To the right of these fields, it shows "Conectado: No" and "Dirección IP: -". There are two buttons at the bottom right: "Reiniciar conexión" and "Guardar".

En modo "PPPoE"

En la página de la red WAN podemos deshabilitar la interfaz, configurarla de modo estático o conectarnos a Internet mediante el protocolo PPPoE.

En el modo estático podremos configurar opcionalmente una IP de pasarela o gateway por defecto, mientras que en el modo PPPoE podremos configurar el nombre de usuario, la contraseña y la VLAN del proveedor de acceso.

Además, una vez configuramos los datos de conexión PPPoE, podremos ver en tiempo real si se establece la conexión y qué dirección IP nos ha asignado nuestro proveedor de acceso. Esto se ha programado utilizando una llamada AJAX a través de la librería JQuery.

Por último, se ha añadido un botón que permite reiniciar la conexión PPPoE en caso de que queramos obtener una nueva IP o algo no vaya bien.

10.5. PÁGINA DE RED LAN

Pi-Router

SISTEMA WAN **LAN** WI-FI RUTAS PUERTOS DMZ LOGS

Red Local

Dirección *: 192.168.1.1

Máscara *: 255.255.255.0

Red: 192.168.1.0/24

Broadcast: 192.168.1.255

Guardar

Servidor DHCP

Habilitar: Si ▾

Dirección Inicial *: 192.168.1.100

Dirección Final *: 192.168.1.150

Tiempo de Renovación: 8 (en horas)

DNS Primario: 1.1.1.1 (opcional)

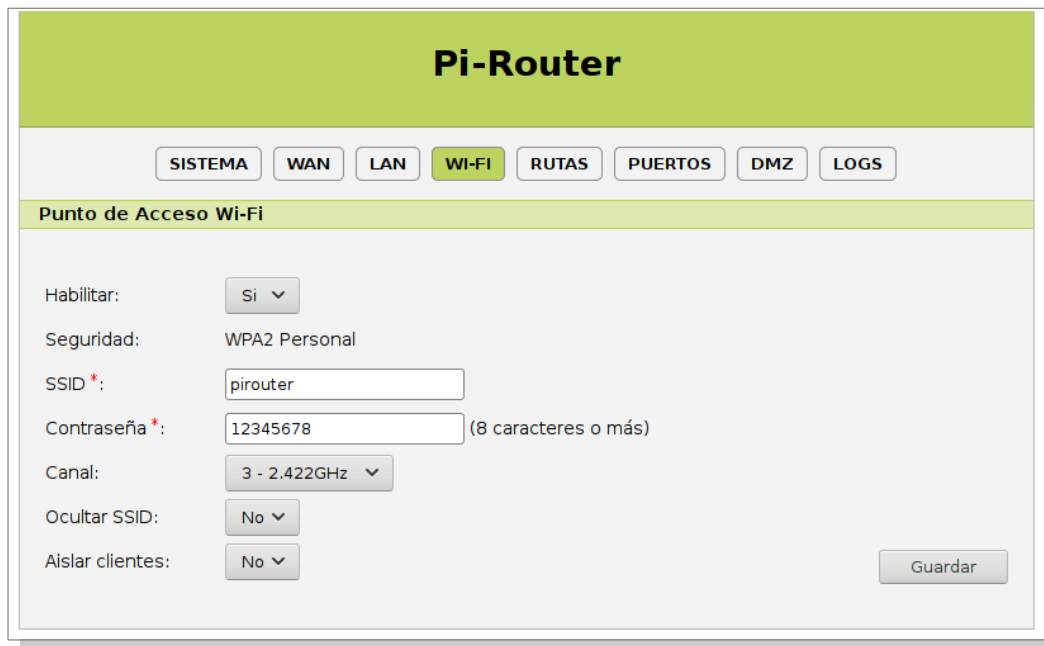
DNS Secundario: 8.8.8.8 (opcional)

Guardar

En esta página podremos configurar la red de área local a través de la IP que le asignaremos al router. A diferencia de los routers comerciales que sólo suelen soportar redes de clase C o B, aquí podemos usar cualquier máscara.

Por otra parte también disponemos de los ajustes del servidor de DHCP. Podemos habilitarlo y deshabilitarlo, establecer el rango de direcciones a repartir así como el tiempo de renovación (LEASE). Los servidores de DNS son opcionales.

10.6. PÁGINA DE AJUSTES WIFI



The screenshot shows the Pi-Router web interface. At the top, there's a green header with the text "Pi-Router". Below it, a navigation bar contains several tabs: SISTEMA, WAN, LAN, WI-FI (which is highlighted in green), RUTAS, PUERTOS, DMZ, and LOGS. Under the "Punto de Acceso Wi-Fi" section, there are several configuration options: "Habilitar:" with a dropdown menu set to "Si"; "Seguridad:" with a text field containing "WPA2 Personal"; "SSID *:" with a text field containing "pirouter"; "Contraseña *:" with a text field containing "12345678" and a note "(8 caracteres o más)"; "Canal:" with a dropdown menu set to "3 - 2.422GHz"; "Ocultar SSID:" with a dropdown menu set to "No"; and "Aislar clientes:" with a dropdown menu set to "No". A "Guardar" button is located at the bottom right of the form.

Aquí podemos habilitar y deshabilitar el punto de acceso Wi-Fi. Como ya se indicó en el capítulo correspondiente, sólo se ha soportado el algoritmo de seguridad WPA-2 Personal.

Podemos modificar el SSID y la contraseña de acceso a la red. También podemos ajustar el canal de transmisión así como ocultar el anuncio de SSID (por motivos de seguridad) y aislar a los clientes WIFI.

Aislar a los clientes WIFI significa que no podrán conectarse a otros clientes WIFI incluso estando en la misma red. Esto es un importante ajuste de seguridad que beneficia a sitios públicos donde los clientes no se conocen entre si.

10.7. PÁGINA DE RUTAS ESTÁTICAS

Pi-Router

SISTEMA WAN LAN WI-FI **RUTAS** PUERTOS DMZ LOGS

Enrutamiento Estático

Importante

- Sólo se pueden añadir rutas estáticas a la interfaz **WAN** en el modo de configuración **Estático**.
- En caso de modificar la **Red** a la que pertenece una interfaz, las rutas estáticas de dicha interfaz son eliminadas automáticamente.

Red Máscara Pasarela Interfaz

Tabla de Rutas

Red	Máscara	Pasarela	Interfaz	Acción
10.20.30.0	255.255.255.0	192.168.1.5	LAN	Eliminar

En esta página podremos añadir las rutas estáticas que necesitemos. Para ello sólo tenemos que indicar la red de destino y su máscara, así como la pasarela o gateway que se va a usar. Hay que indicar la interfaz de red a la que está conectada la pasarela.

En la imagen se puede ver como se ha añadido una ruta de prueba a la red “10.20.30.0/24” con pasarela “192.168.1.5” a través de la interfaz LAN.

Las rutas estáticas de la interfaz WAN sólo se aplican cuando está configurada en modo estático.

10.8. PÁGINA DE REENVÍO DE PUERTOS

The screenshot shows the Pi-Router web interface. At the top, there's a green header with 'Pi-Router' in white. Below it, a navigation bar contains buttons for 'SISTEMA', 'WAN', 'LAN', 'WI-FI', 'RUTAS', 'PUERTOS' (highlighted in green), 'DMZ', and 'LOGS'. The main content area is titled 'Reenvío de Puertos'. Under this, there's a section 'Importante' with a bullet point: '• Sólo aplicable en modo Gateway.' Below this is a form to add a new rule with fields for 'Regla' (empty), 'Protocolo' (dropdown menu showing 'TCP'), 'Puerto Inicial' (empty), 'Puerto Final' (empty), and 'Equipo' (empty). There's an 'Añadir' button below the form. At the bottom, there's a table titled 'Tabla de Reenvío de Puertos' with the following data:

Regla	Protocolo	Puerto Inicial	Puerto Final	Equipo	Acción
Servidor WEB	tcp	80	80	192.168.1.20	Eliminar

Esta página es similar a la de rutas estáticas, pero ahora podremos añadir el reenvío de uno o más puertos a un equipo de la red. Podemos indicar el nombre de la regla, el protocolo, un rango de puertos y la IP del equipo en cuestión.

En la imagen vemos una regla para reenviar el puerto 80 del router al puerto 80 del servidor web situado en “192.168.1.20”.

El reenvío de puertos sólo se activa cuando el router trabaja en modo “Gateway”.

10.9. PÁGINA DE HOST DMZ



The screenshot shows the Pi-Router web interface. At the top is a green header with the text "Pi-Router". Below the header is a navigation bar with buttons for "SISTEMA", "WAN", "LAN", "WI-FI", "RUTAS", "PUERTOS", "DMZ", and "LOGS". The "DMZ" button is highlighted in green. Below the navigation bar is a sub-header "Host DMZ" in a light green bar. The main content area has a section titled "Importante" with a bullet point: "• Sólo aplicable en modo **Gateway**." Below this is a form field labeled "Host:" followed by a text input box and the text "(opcional)". To the right of the input box is a "Guardar" button.

Esta página es muy sencilla, tan solo tenemos que indicar la IP de un equipo de la red para asignarlo como host DMZ. De esta forma todo el tráfico ira a el por defecto. Para eliminarlo tan solo tenemos que dejar el campo en blanco.

El host DMZ sólo se activa en el modo de trabajo “*Gateway*”.

10.10. PÁGINA DE REGISTROS

Pi-Router

SISTEMA
WAN
LAN
WI-FI
RUTAS
PUERTOS
DMZ
LOGS

Registro de Aplicación

```

192.168.1.2 - - [22/Apr/2020:11:01:27 BST] "POST /add_port HTTP/1.1" 303 0
http://192.168.1.1/ports -> /add_port
192.168.1.2 - admin [22/Apr/2020:11:01:27 +0100] "GET /ports HTTP/1.1" 200 2592 0.0068
192.168.1.2 - - [22/Apr/2020:11:01:27 BST] "GET /ports HTTP/1.1" 200 2592
http://192.168.1.1/ports -> /ports
192.168.1.2 - admin [22/Apr/2020:11:01:28 +0100] "GET /css/main.css HTTP/1.1" 304 - 0.0041
192.168.1.2 - - [22/Apr/2020:11:01:28 BST] "GET /css/main.css HTTP/1.1" 304 0
http://192.168.1.1/ports -> /css/main.css
192.168.1.2 - admin [22/Apr/2020:11:01:28 +0100] "GET /js/jquery-3.4.1.min.js HTTP/1.1" 304 - 0.0051
192.168.1.2 - - [22/Apr/2020:11:01:28 BST] "GET /js/jquery-3.4.1.min.js HTTP/1.1" 304 0
http://192.168.1.1/ports -> /js/jquery-3.4.1.min.js
192.168.1.2 - admin [22/Apr/2020:11:01:41 +0100] "GET /dmz HTTP/1.1" 200 1542 0.0206
192.168.1.2 - - [22/Apr/2020:11:01:41 BST] "GET /dmz HTTP/1.1" 200 1542
http://192.168.1.1/ports -> /dmz
192.168.1.2 - admin [22/Apr/2020:11:01:41 +0100] "GET /css/main.css HTTP/1.1" 304 - 0.0042
192.168.1.2 - admin [22/Apr/2020:11:01:41 +0100] "GET /js/jquery-3.4.1.min.js HTTP/1.1" 304 - 0.0037
192.168.1.2 - - [22/Apr/2020:11:01:41 BST] "GET /css/main.css HTTP/1.1" 304 0
http://192.168.1.1/dmz -> /css/main.css
192.168.1.2 - - [22/Apr/2020:11:01:41 BST] "GET /js/jquery-3.4.1.min.js HTTP/1.1" 304 0
http://192.168.1.1/dmz -> /js/jquery-3.4.1.min.js

```

Registro de Sistema

```

-- Logs begin at Wed 2020-04-22 10:26:49 BST, end at Wed 2020-04-22 11:01:23 BST. --
Apr 22 10:28:20 raspberrypi2 dnsmasq-dhcp[626]: DHCPOFFER(bridge0) 192.168.1.139 56:dd:1c:00:9c:e1
Apr 22 10:32:03 raspberrypi2 dnsmasq-dhcp[626]: DHCPDISCOVER(bridge0) 56:dd:1c:00:9c:e1
Apr 22 10:32:03 raspberrypi2 dnsmasq-dhcp[626]: DHCPOFFER(bridge0) 192.168.1.139 56:dd:1c:00:9c:e1
Apr 22 10:42:00 raspberrypi2 systemd[1]: Starting Cleanup of Temporary Directories...
Apr 22 10:42:00 raspberrypi2 systemd[1]: systemd-tmpfiles-clean.service: Succeeded.
Apr 22 10:42:00 raspberrypi2 systemd[1]: Started Cleanup of Temporary Directories.
Apr 22 10:59:40 raspberrypi2 avahi-daemon[319]: Joining mDNS multicast group on interface
enx74da38ff9ab5.IPv4 with address 172.26.0.7.
Apr 22 10:59:40 raspberrypi2 avahi-daemon[319]: New relevant interface enx74da38ff9ab5.IPv4 for mDNS.
Apr 22 10:59:40 raspberrypi2 avahi-daemon[319]: Registering new address record for 172.26.0.7 on
enx74da38ff9ab5.IPv4.
Apr 22 10:59:53 raspberrypi2 avahi-daemon[319]: Withdrawing address record for 172.26.0.7 on
enx74da38ff9ab5.
Apr 22 10:59:53 raspberrypi2 avahi-daemon[319]: Leaving mDNS multicast group on interface
enx74da38ff9ab5.IPv4 with address 172.26.0.7.
Apr 22 10:59:53 raspberrypi2 avahi-daemon[319]: Interface enx74da38ff9ab5.IPv4 no longer relevant for mDNS.

```

Aquí podemos ver tanto los registros del servidor web de la aplicación como los registros del sistema (**journalctl**). Muy útil para descubrir si algo va mal en el router.

11. BIBLIOGRAFÍA

Instalación de Raspbian 10

<https://www.raspberrypi.org/documentation/installation/installing-images/>

Nombres Predictivos de Interfaces de Red

<https://wiki.debian.org/NetworkInterfaceNames>

Simulador de Dispositivos de Red de Linksys

<https://ui.linksys.com/>

Algoritmos de Seguridad WIFI

<https://www.howtogeek.com/204697/wi-fi-security-should-you-use-wpa2-aes-wpa2-tpk-tpk-or-both/>

Bridge en Arch Linux

https://wiki.archlinux.org/index.php/Network_bridge

Bridge en Debian Linux

<https://wiki.debian.org/BridgeNetworkConnections>

Punto de Acceso WIFI con Raspberry Pi

<https://thepi.io/how-to-use-your-raspberry-pi-as-a-wireless-access-point/>

Fragmentación en Ipv4. MTU, MSS y PMTU.

<https://www.cisco.com/c/en/us/support/docs/ip/generic-routing-encapsulation-gre/25885-pmtud-ipfrag.html>

MSS Clamping en PPPoE

<https://lartc.org/howto/lartc.cookbook.mtu-mss.html>

Identificadores VLAN de operadores de fibra

https://wiki.bandaancho.st/Identificadores_VLAN_operadores_FTTH

Configuración de VLAN para operadores de fibra

<https://internetpasoapaso.com/configurar-vlan/>

PPPoE en Ubuntu Linux

<https://help.ubuntu.com/community/ADSLPPPoE>

PPPoE en Wikipedia

https://en.wikipedia.org/wiki/Point-to-Point_Protocol_over_Ethernet

Reenvío de Puertos con IPTables

<https://www.systutorials.com/port-forwarding-using-iptables/>

Tutorial de DNSMASQ

<https://www.linux.com/training-tutorials/dnsmasq-easy-lan-name-services/>

Configurar zona DMZ en Linux

<https://www.milesweb.com/blog/technology-hub/how-to-set-up-a-dmz-with-linux/>

Documentación del microframework Sinatra

<http://sinatrarb.com/intro.html>

Autenticación HTTP en Sinatra

<http://sinatrarb.com/faq.html#auth>

Librería Estandar de Ruby

<https://ruby-doc.org/stdlib/>

Ajax con JQuery

<https://api.jquery.com/jquery.ajax/>

NAT HowTo

<https://www.netfilter.org/documentation/HOWTO/NAT-HOWTO-6.html>