



## 2.1.3

### ***1. Investigar cómo configurar el servidor DNS local en el sistema operativo donde estén trabajando. ¿Qué pasos deben seguirse para tal fin?***

Para este punto, explicaremos como lo hicimos nosotros que somos usuarios de Windows 10.

1. Primero entramos al menú de configuración de Windows10 y luego vamos a la sección de configuración de Red.
2. Luego nos vamos a la configuración Wifi/Ethernet dependiendo como tengamos conectada nuestra PC (en nuestro caso WIFI). Aquí seleccionamos la opción de modificar las opciones del adaptador.
3. En este panel, veremos nuestra conexión actual. Vamos a entrar en la configuración de propiedades de esta.
4. El siguiente paso es apretar en IPV4 y dirigirnos a sus propiedades.
5. Aquí seleccionamos que queremos usar la dirección 127.0.0.1 para resolver DNS.

De por si estos pasos no cambian el servidor DNS local del sistema operativo, pero si el de la red que en definitiva tiene el mismo efecto mientras no se cambie de red.

### ***2. ¿Qué significado tiene la dirección IP 127.0.0.1?***

La IP 127.0.0.1 hace referencia al localhost, lo cual significa que esta IP es la de nuestro equipo.

### ***3. A partir de todo lo anterior, ejecutar el proxy DNS y colocar la dirección IP 127.0.0.1 como servidor DNS local del sistema operativo. Abrir un navegador web a elección y navegar a [www.google.com](http://www.google.com). ¿Qué se observa en la consola del proxy DNS? ¿El navegador web logró mostrar la página de inicio de Google?***

```
[*]Query recibida: A www.google.com (de 127.0.0.1:63032)
[*]Respondiendo 142.251.133.228 (vía 1.1.1.1)
```

Si, se logró mostrar.

### ***4. Dejar que transcurra cierto tiempo hasta que aparezca algún otro log en la consola del proxy DNS. ¿A qué pueden deberse estas queries?***

Puede surgir por aplicaciones en segundo plano, por ejemplo, si estamos usando Spotify y pasa de canción, Spotify realizara una query automáticamente para descargar la misma.

### ***5. Repetir la pregunta 3. ejecutando el proxy DNS con un mapeo predeterminado de [www.google.com](http://www.google.com) a la dirección IP de [www.infobae.com](http://www.infobae.com). ¿Qué sucede en este caso? ¿Por qué?***

Recibimos la siguiente salida por la consola:

```
[*]Query recibida: A www.google.com (de 127.0.0.1:64966)
[*]Respondiendo 95.101.24.158 (predeterminado)
```

Si hacemos un dig de [www.google.com](http://www.google.com) nos devuelve en la zona de respuestas que recibió la IP 190.94.188.51 que es la que le corresponde a Infobae. Sin embargo, cuando lo corremos en el navegador Firefox, devuelve una advertencia que dice que no se pudo conectar debido a un posible riesgo de seguridad. Devuelve esto porque, como explica la advertencia, el certificado de seguridad que estaba esperando no le llego y por eso no puede abrir la página.

## 2.2.3

**1. Ejecutar el servidor HTTP y en otra consola utilizar *netcat* para conectarse al mismo y reproducir el tercer ejemplo de la sección 2.2.2. Transcribir los intercambios de mensajes HTTP. ¿Se obtuvo el funcionamiento esperado?**

La consola de netcat: (en azul son inputs nuestros)

```
GET / HTTP/1.1 Host: www.infobae.com
HTTP/1.1 200 OK
Content-Type: text/html
<html>
  <head>
    <title>HTTP 200 Response</title>
  </head>
  <body>
    <h1>Success!</h1>
    <p>This is an example of an HTTP response with a status code of 200
and HTML content.</p>
  </body>
</html>

GET / HTTP/1.1 Host: www.uba.ar
HTTP/1.1 301 Moved Permanently
Location: https://www.utdt.edu

GET / HTTP/1.1 Host: www.google.com
HTTP/1.1 301 Moved Permanently
Location: https://google.com
```

Si se obtuvo el comportamiento esperado. En el tercero, el de Google, devolvemos la dirección sin el “www.”, porque después nos causó problemas en el tercer punto.

**2. Intentar reproducir alguno de estos ejemplos desde un navegador web. ¿Por qué no es posible hacerlo?**

No es posible porque el navegador no usa como resolvedor a nuestro servidor.

**3. Pensar y ejecutar una manera de interactuar con el servidor HTTP desde un navegador web. ¿Qué URL deberíamos utilizar en el navegador?**

Nosotros lo que hicimos fue configurar Firefox para que use el servidor local y puerto 80 en la sección de Proxy HTTP.

Deberíamos usar el URL de HTTP de las páginas.

## 2.3. Tercera parte: integración

*1. Como hicimos en la primera parte, configurar el sistema operativo local de modo tal que las consultas DNS se resuelvan a través de nuestro proxy DNS. Ejecutar ambos servidores y reconsiderar nuevamente la pregunta 2 de la sección 2.2.3: ¿Cómo podemos capitalizar la funcionalidad del proxy DNS para poder interactuar con el redirector HTTP desde un navegador web utilizando dominios arbitrarios?*

Para lograr que interactúen el servidor DNS con el de HTTP para un sitio web X, deberíamos en el servidor DNS cuando lo corremos por consola inicializarlo con “-d X:Y ” con Y igual la dirección IP del servidor HTTP.

*2. En el contexto de una LAN a elección, configurar un host h para que resuelva consultas DNS a través de otro host h’ donde esté en ejecución nuestro proxy DNS y nuestro redirector HTTP. Abrir un navegador web en h e intentar recrear alguno de los ejemplos del punto anterior. ¿Se obtuvo el funcionamiento esperado?*

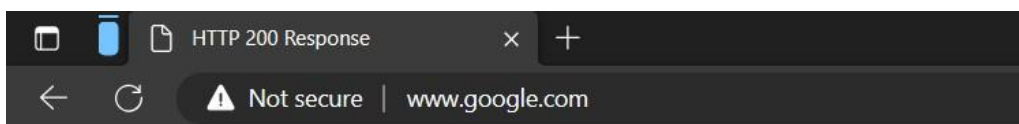
Intentamos el siguiente ejemplo:

```
$ sudo python3 http_redirector.py -c content.html -d www.infobae.com -d www.google.com
[*] Request GET recibido (Host: www.infobae.com)
[*] Respondiendo contenido del archivo content.html
[*] Request GET recibido (Host: www.google.com)
[*] Respondiendo contenido del archivo content.html
```

Se obtuvo el comportamiento esperado para *www.google.com*, pero para *www.infobae.com* no. Esto se debió a que todos los navegadores que probamos mandan los paquetes de *Infobae* al puerto 443 y no logramos encontrar manera de forzar que lo mande al puerto 80.

```
[*]Respondiendo b'www.tm.ak.prd.aadg.akadns.net.' (vía 1.1.1.1)
[*]Query recibida: A www.google.com (de 192.168.3.21:53308)
[*]Respondiendo 192.168.3.32 (predeterminado)
[*]Query recibida: HTTPS www.google.com (de 192.168.3.21:61773)
[*]Respondiendo b'\x00\x01\x00\x01\x00\x06\x02h2\x02h3' (vía 1.1.1.1)
[*]Query recibida: A login.microsoftonline.com (de 192.168.3.21:63727)
[*]Respondiendo 40.126.45.20 (vía 1.1.1.1)
[*]Query recibida: HTTPS login.microsoftonline.com (de 192.168.3.21:594)
[*]Respondiendo b'www.tm.ak.prd.aadg.akadns.net.' (vía 1.1.1.1)
```

```
PS C:\Users\Usuario\Desktop\Td4-main\TD> & C:/Users/Usuario/AppData/Local/Programs/Python/Python311/python.exe C:\Users\Usuario\Desktop\Td4-main\TD\http_redirector.py -c content.html -d www.google.com -d www.infobae.com
Hay servidor
Request GET recibido (Host: www.google.com)
[*] Respondiendo contenido del archivo content.html
Request GET recibido (Host: www.google.com)
[*] Respondiendo contenido del archivo content.html
```



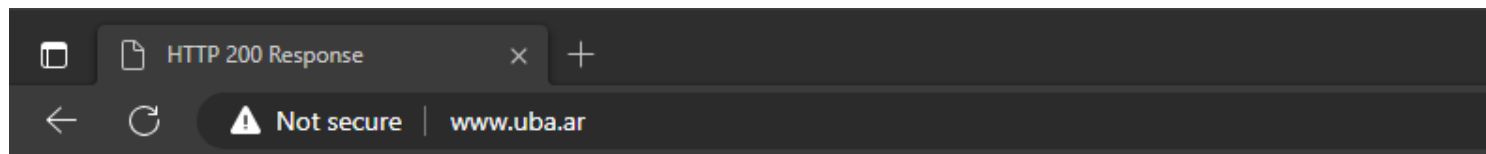
### Success!

This is an example of an HTTP response with a status code of 200 and HTML content.

*3. Reconsiderar el experimento del punto anterior ejecutando el proxy DNS en un host h’ de la LAN y el redirector HTTP en otro host h’’ distinto. ¿Se obtuvo el funcionamiento esperado?*

Si se obtuvo el funcionamiento esperado, ya que se replicó lo mismo que en el punto anterior.

Estas son unas screenshots que tomamos mientras hacíamos el punto 2:



## Success!

This is an example of an HTTP response with a status code of 200 and HTML content.

