



**PONTIFICIA UNIVERSIDAD JAVERIANA**  
**FACULTAD DE INGENIERIA**  
**DEPARTAMENTO DE INGENIERIA DE SISTEMAS**  
Comunicaciones y Redes  
Período Académico 2021-30

**Primer Proyecto – Entrega final lunes 25 de octubre, Sustentación: Lunes 25 y miércoles 27 de octubre**

### **Objetivo.**

El servidor se tiene que desarrollar en C/C++ o Java, se puede utilizar cualquier otro lenguaje de **programación que todos los integrantes del grupo conozcan** y en el que puedan hacer modificaciones de manera individual en el momento de la sustentación. **Se autoriza el uso de clases o librerías que faciliten el uso del protocolo HTTP.**

**Para realizar su papel de Proxy Web, el programa debe funcionar simultáneamente como un servidor Web que recibe solicitudes de los navegadores y como un cliente web al reenviar estas solicitudes al servidor de destino original, por lo tanto, el objetivo del proyecto es desarrollar un servidor Proxy Web** (que maneje el protocolo HTTP 1.0), en donde el servidor proxy actúa como intermediario, recibiendo las solicitudes de clientes que deben ser navegadores comerciales, como Chrome y Microsoft Edge entre otros, estas solicitudes serán reenviadas al sitio web de destino y la respuesta del servidor web debe ser reenviada al navegador.

El servidor proxy debe tener la funcionalidad de “*sitios web virtuales*”, por lo tanto, si el sitio web especificado en la solicitud corresponde a un “*sitio web virtual*” se debe tener una tabla en que se tenga la correspondencia entre el nombre virtual del sitio y el host donde verdaderamente se encuentra la información. En este caso, el proxy web debe modificar la solicitud como se describe más adelante. Si el sitio web NO es un “*sitio web virtual*” la solicitud se reenvía sin modificaciones al sitio especificado.

El servidor debe mostrar en tiempo real las solicitudes recibidas y el reenvío correspondiente a esta solicitud, también se debe generar un log con estas transacciones en un archivo texto. La información mostrada en pantalla y en el archivo de log, debe indicar los campos más importantes del mensaje y el tratamiento que se le da, por ejemplo si la solicitud corresponde a un “*sitio web virtual*” se debe indicar el URL al que se reenviará.

El programa debe manejar los dos tipos de solicitudes (GET y POST) que se describen a continuación, para comprender el funcionamiento del protocolo, el grupo debe leer y entender el o los RFCs correspondientes y describirlos en el documento que se entrega junto al programa.

### **Descripción de las solicitudes de tipo GET.**

A continuación se transcribe **un ejemplo** de solicitud de tipo GET enviada por un navegador, con el fin de identificar las partes más relevantes. La solicitud la componen una serie de líneas de texto terminadas por un salto de línea (los caracteres Carriage Return, Line Feed; que no son visibles en esta transcripción).

```
GET http://www.javeriana.edu.co/home-visitantes/noticias/galeria/random/4.swf HTTP/1.0
Referer: http://www.javeriana.edu.co/home-visitantes/noticias/noticias_flash.html
Proxy-Connection: Keep-Alive
User-Agent: Mozilla/4.76 [en] (Windows NT 5.0; U)
Pragma: no-cache
Host: www.javeriana.edu.co
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

En la primera línea se identifica el tipo de solicitud (en este caso es de tipo `GET`), el URL del recurso que se está solicitando, así como la versión del protocolo utilizada (`HTTP/1.0`).

A continuación viene una serie de encabezados que dan información adicional sobre la solicitud y que terminan con una línea en blanco (que no es visible en esta transcripción).

El único encabezado que debe analizar el servidor proxy web es el atributo `Host` que especifica el servidor web al que hay que conectarse y reenviar la solicitud.

En nuestro caso este nombre de host puede corresponder a un nombre de “sitio web virtual” o a un host real. Si se trata de un “sitio virtual” el procesamiento se describe más adelante. En el caso del sitio real la solicitud se envía sin modificaciones al servidor especificado.

### Descripción de las solicitudes de tipo `POST`.

El otro tipo de solicitud que debe manejar el servidor proxy web son las solicitudes de tipo `POST`, que se utilizan para enviar datos desde el browser hacia el servidor, por ejemplo, en el caso en que el usuario complete una forma HTML. A continuación se transcribe **un ejemplo** de solicitud de tipo `POST` enviada por un browser, con el fin de identificar las partes más relevantes.

La solicitud la componen una serie de líneas de texto terminadas por un salto de línea (los caracteres `Carriage Return`, `Line Feed` que no son visibles en esta transcripción).

```
POST http://sophia.javeriana.edu.co/scripts/login.php3 HTTP/1.0
Referer: http://sophia.javeriana.edu.co/
Proxy-Connection: Keep-Alive
User-Agent: Mozilla/4.76 [en] (Windows NT 5.0; U)
Host: ainsuca.javeriana.edu.co
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
Content-type: application/x-www-form-urlencoded
Content-length: 60
```

```
f_login=usuario&f_clave=pwd12345&imageField.x=31&imageField.y=14
```

En la primera línea se identifica el tipo de solicitud (en este caso es de tipo `POST`), el URL del recurso que se está solicitando, así como la versión del protocolo utilizada (`HTTP/1.0`).

A continuación viene una serie de encabezados que dan información adicional sobre la solicitud y que terminan con una línea en blanco, después viene el cuerpo del mensaje donde se envían los datos que el usuario ingresó en la forma.

La diferencia principal con la solicitud `GET` es que después de los encabezados vienen datos que deben ser leídos por el proxy web y reenviados al servidor especificado (virtual o real).

Al igual que con la solicitud `GET` el servidor proxy web debe analizar el atributo `Host` para determinar el servidor web al que hay que conectarse y reenviar la solicitud. Adicionalmente, en el caso de una solicitud `POST`, se debe analizar el atributo `Content-length` que especifica el tamaño de los datos (en bytes) que conforman el cuerpo de la solicitud, esto le permite al proxy saber donde termina la solicitud.

### Descripción de la respuesta del servidor.

Independientemente de la solicitud enviada las respuestas de los servidores web tienen el mismo formato. Estos datos deben ser enviados al browser que los solicitó sin ninguna modificación. A continuación, se transcribe **un ejemplo** de una respuesta enviada por un servidor web, con el fin de identificar las partes más relevantes. La respuesta la componen una serie de líneas de texto terminadas por un salto de línea (los caracteres `Carriage Return`, `Line Feed` que no son visibles en esta transcripción).

```
HTTP/1.1 200 OK
Date: Fri, 23 Oct 2015 09:15:30 GMT
Server: Apache/1.3.19 Ben-SSL/1.44 (Unix) PHP/4.0.5
Last-Modified: Fri, 04 Aug 2015 15:54:49 GMT
ETag: "4cee-43cb-398ae749"
Accept-Ranges: bytes
Content-Length: 17355
Connection: close
Content-Type: application/x-shockwave-flash
```

En la primera línea se identifica la versión del protocolo utilizada (en este caso el servidor respondió HTTP/1.1, sin embargo, esto no afecta el funcionamiento del servidor que se va a desarrollar porque estos datos no se analizan) y el código de éxito o fracaso de la operación.

A continuación viene una serie de encabezados que dan información adicional sobre la respuesta y que terminan con una línea en blanco (que no es visible en esta transcripción). Y después viene el cuerpo del mensaje (que en este ejemplo no se muestran).

El único encabezado de la respuesta que se debe analizar es el atributo `Content-length`, que especifica el tamaño de los datos (en bytes) que conforman el cuerpo de la respuesta. Después de leer los encabezados y los datos solicitados el proxy web puede cerrar el socket con el servidor web y con el browser y dar por terminada la solicitud.

### Manejo de “sitios web virtuales”.

Para implementar la funcionalidad de “*sitios web virtuales*” el servidor proxy web debe tener un archivo de configuración en el que se asocie un nombre de host virtual con el host real donde se encuentra y el directorio de base en ese host.

Por ejemplo, la tabla de host virtuales puede ser la siguiente:

Nombre Host virtual	Host real	Directorio raíz
r.paez	sophia.javeriana.edu.co	~rpaez/
e.ruiz	sophia.javeriana.edu.co	~eruiz/

Cuando se recibe una solicitud (GET o POST) se debe verificar el host solicitado (especificado en el encabezado `Host`) y compararlo con los nombres de host virtuales configurados.

Si el host especificado es un host virtual se deben hacer las siguientes modificaciones a la solicitud antes de enviarla:

1. Modificar el URL solicitado: Se debe cambiar la primera línea de la solicitud cambiando el host del URL por el host real y el directorio raíz que se configuraron.
2. Modificar el encabezado `Host`: Se debe cambiar la línea correspondiente al encabezado `Host` por una que contenga el nombre del host real.

**Por ejemplo**, si la solicitud es la siguiente (en negrilla los campos que se deben modificar):

```
GET http://e.ruiz/index.html HTTP/1.0
Proxy-Connection: Keep-Alive
User-Agent: Mozilla/4.76 [en] (Windows NT 5.0; U)
Pragma: no-cache
Host: e.ruiz
Accept: */*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

Se debe convertir en (en negrilla los campos modificados):

```
GET http://sophia.javeriana.edu.co/~eruiz/index.html HTTP/1.0
Proxy-Connection: Keep-Alive
User-Agent: Mozilla/4.76 [en] (Windows NT 5.0; U)
Pragma: no-cache
Host: sophia.javeriana.edu.co
Accept: */*
```

Accept-Language: en  
Accept-Charset: iso-8859-1,\*,utf-8

Las solicitudes POST se manejan de forma idéntica. Si el host solicitado no se encuentra en la lista de host virtuales se debe suponer que se trata de un host real y conectarse directamente a él, enviándole la solicitud sin ninguna modificación.

### **Equipos de trabajo**

El proyecto se realizará en grupos de máximo 4 personas.

Se debe notificar al profesor quienes son los integrantes del grupo a la entrega de este enunciado y tendrán 15 días para que los integrantes puedan realizar cambios de grupo.

### **Entrega y condiciones**

La entrega del proyecto se realizará a través de la plataforma BrightSpace el **miércoles 25 de octubre**, a más tardar en la hora de inicio de la clase. El proyecto se debe subir a la plataforma en un solo archivo comprimido.

Parte de la entrega consiste en un documento que explique el funcionamiento del protocolo HTTP, el diseño de programa, así como el escenario de pruebas, las pruebas realizadas y sus resultados. Se debe utilizar la plantilla para la elaboración de artículos de la IEEE y la extensión del documento no debe ser superior a 8 páginas; también se debe hacer un vídeo de máximo 5 minutos que muestre el funcionamiento del proyecto. **En ningún caso se considera documentación al código fuente.**

La sustentación se realizará el **miércoles 25 y viernes 27 de octubre** en el horario de la clase, según hora definida para cada grupo, la sustentación tendrá una duración de una hora; **el grupo debe asegurar que su proyecto funciona antes de iniciar la sustentación, no se dará tiempo para estos ajustes.**

**Los integrantes del grupo que no estén presentes en el momento de iniciar la sustentación obtendrán como nota definitiva del proyecto Cero punto Cero (0.0), La sustentación iniciará puntualmente en la hora seleccionada por el grupo, solo podrán sustentar las personas que se encuentren presentes en ese momento. Una vez iniciada la sustentación no se aceptará el ingreso de ninguna otra persona al grupo y se deberá seguir el reglamento estudiantil para la solicitud de una prueba supletoria.**

**La nota de la sustentación es individual, una sustentación no satisfactoria causa una nota de 0.0 en el proyecto, así éste funcione correctamente.**