

# Informe del Trabajo Práctico N.º 6

## Parte A:

### Protocolo HTTP. Servidores y aplicaciones web

#### Introducción

El presente informe detalla el desarrollo del Trabajo Práctico N.º 6 – Parte A de la materia Redes de Computadoras, cuyo objetivo principal fue instalar y configurar un servidor web Apache, y desarrollar una aplicación web sencilla que integre tecnologías del lado del cliente (HTML, CSS, JavaScript) y del lado del servidor (PHP). Asimismo, se realizaron análisis de tráfico de red mediante Wireshark para observar el comportamiento del protocolo HTTP y el intercambio de paquetes durante el funcionamiento de la aplicación. Este trabajo permitió afianzar conceptos teóricos sobre protocolos de red, servidores web y aplicaciones dinámicas, mediante la implementación práctica de una solución funcional y su análisis desde la capa de red.

#### Actividad 1

Para cumplir con los objetivos de la Actividad 1 del Trabajo Práctico N.º 6, se instaló y configuró correctamente el servidor Apache, ejecutándose sobre un entorno Linux. La aplicación web desarrollada consiste en una encuesta que permite al usuario seleccionar su estación del año favorita e ingresar un correo electrónico para validar su voto.

#### Estructura y funcionamiento del sitio:

El sitio web está compuesto por tres archivos principales:

- `index.html`: Página principal que contiene la estructura del formulario HTML y la lógica de validación con JavaScript.
- `style.css`: Hoja de estilos utilizada para dar formato visual a los elementos del formulario y las imágenes.
- `verificar.php`: Script PHP que procesa los datos del formulario, registra los votos en archivos de texto y muestra los resultados de la encuesta.

#### Componentes del formulario:

En `index.html` se encuentra:

- **Un campo para ingresar e-mail:** Se valida que tenga al menos 7 caracteres, que contenga un `@` y un `.` válidos, que no estén al inicio ni al final, y que no incluya caracteres especiales como `#`, `$`, `!`, `%`, etc.
- **Opciones tipo radio** para seleccionar una estación favorita (Otoño, Invierno, Primavera o Verano), acompañadas por imágenes representativas

Integrantes: Martina Nahman y Emiliano Germani

- **Un botón de envío**, estilizado con CSS, que al ser presionado ejecuta una función `validar()` en JavaScript para verificar los datos antes de enviar el formulario.

### Validación en el cliente:

La función `validar()` implementada en JavaScript comprueba:

- La validez del formato del correo electrónico.
- Que el usuario haya seleccionado alguna opción.

Si la validación falla, se muestra un mensaje de alerta detallando los errores.

### Procesamiento del lado del servidor:

Cuando el formulario pasa la validación y se envía, el archivo `verificar.php`:

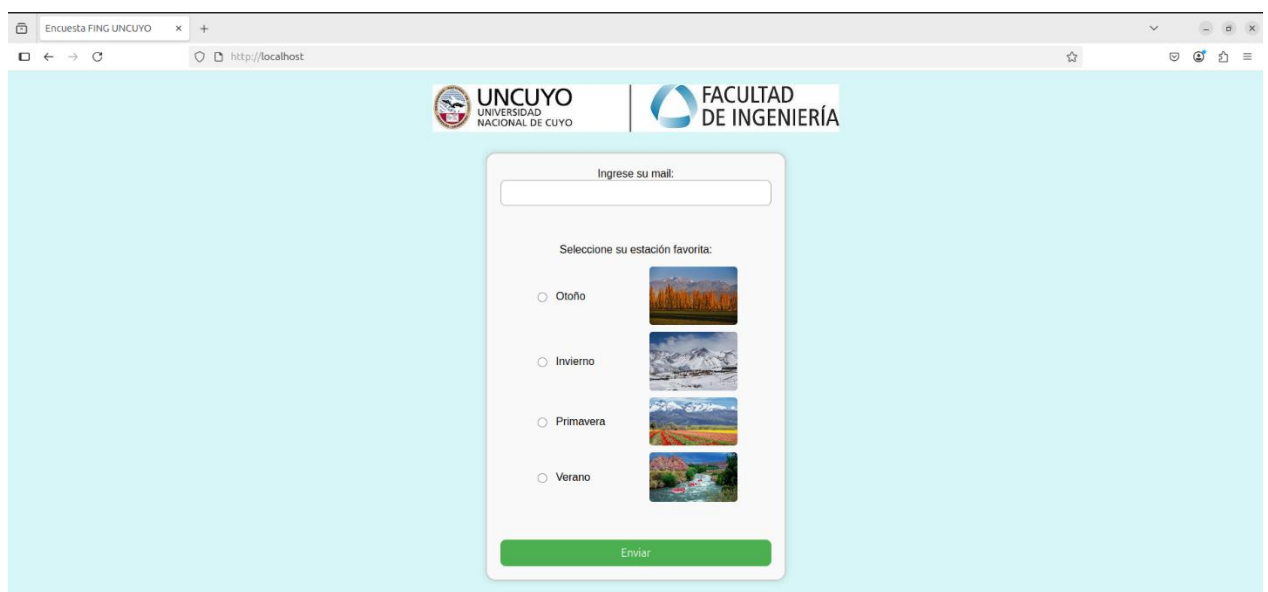
1. **Verifica si el e-mail ya votó:** lee un archivo `votos.txt` donde se registran los votos previos. Si el e-mail ya existe, se muestra un mensaje de que no se puede votar dos veces.
2. **Registra el nuevo voto** en `votos.txt` si el e-mail no había participado.
3. **Actualiza las estadísticas** en `estadisticas.txt`, sumando un voto a la opción correspondiente.
4. **Muestra los resultados acumulados** de la encuesta en pantalla.
5. **Incluye un botón** para volver a la página principal y permitir que otro usuario vote desde el mismo dispositivo.

### Estética y estilo

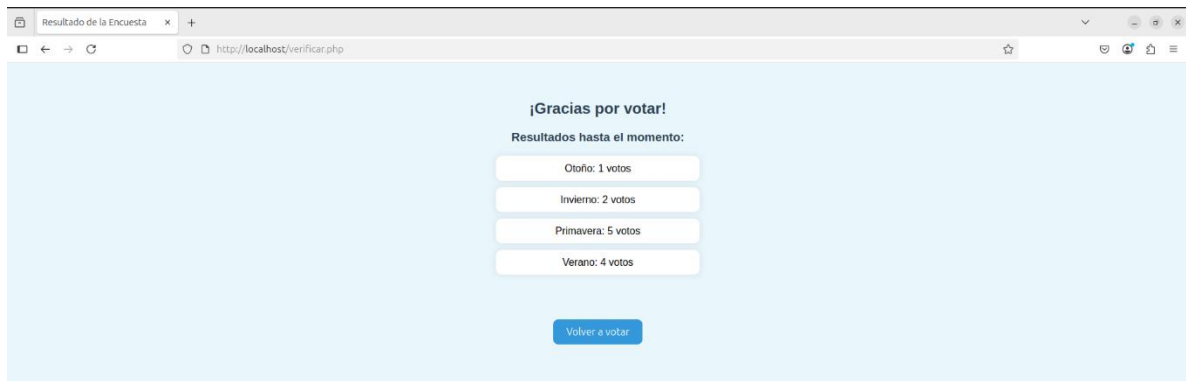
El archivo `style.css` proporciona una apariencia amigable y moderna. Se aplicaron estilos para centrar el formulario, estilizar los inputs y aplicar efectos suaves al pasar el cursor sobre los botones e imágenes. Las imágenes de las estaciones hacen más visual e intuitiva la selección.

### Capturas de pantalla:

A continuación, se incluyen capturas de la aplicación web desarrollada:



Integrantes: Martina Nahman y Emiliano Germani



Quedan registrados los votos en el archivo estadísticas.txt y los correos correspondientes en el archivo votos.txt

```
emiliano@emiliano-notebookAcer: ~  
emiliano@emiliano-notebookAcer:~$ cat /var/www/html/votos.txt  
palermo9@gmail.com;Primavera  
mmmmmmmm@gmail.com;Verano  
mmmmmm8mm@gmail.com;Verano  
holacapoooo@gmail.com;Primavera  
99palermo9@gmail.com;Invierno  
99palermddd9@gmail.com;Primavera  
ejemplo@gmail.com;Otoño  
holaholahola@gmail.com;Verano  
pablo@hola.com;Verano  
holanda@gmail.com;Primavera  
palermocapo9@hotmail.com;Invierno  
prueba@hotmail.com  
emiliano@emiliano-notebookAcer:~$ cat /var/www/html/estadisticas.txt  
Otoño;1  
Invierno;2  
Primavera;5  
Verano;4  
emiliano@emiliano-notebookAcer:~$
```

## Actividad 2 – Análisis del tráfico HTTP

Para esta actividad se utilizó la herramienta **Wireshark** con el objetivo de observar el tráfico HTTP generado durante el uso de la aplicación web desarrollada. Se realizaron las siguientes acciones desde un segundo dispositivo conectado a la misma red que el servidor:

1. Se accedió a la página principal (`index.html`) del sitio web desde un navegador.
2. Se completó el formulario con un e-mail válido y una opción seleccionada, y se presionó el botón “Enviar”.

Durante ese proceso, se identificaron y capturaron cuatro paquetes HTTP relevantes:

### a. Petición GET inicial

Al ingresar a la página, el navegador envió una **petición HTTP GET** solicitando el archivo `index.html`. Esta petición permitió cargar la interfaz web desde el servidor Apache.

## b. Respuesta del servidor a la GET

El servidor respondió con el código 200 OK y devolvió el contenido del archivo HTML al cliente, incluyendo las referencias al CSS y al script JavaScript embebido

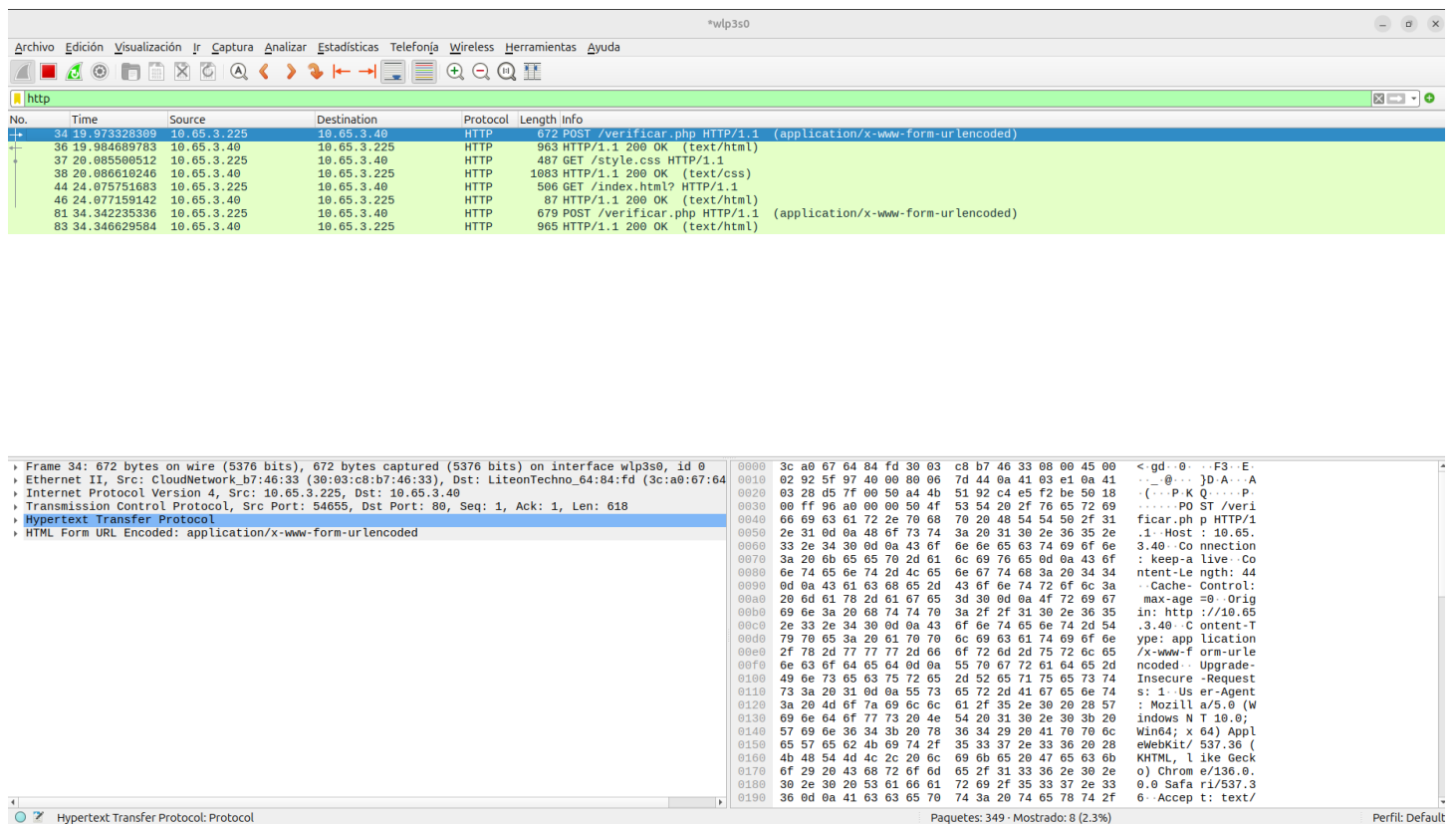
## c. Petición POST del formulario

Al enviar el formulario, se generó una **petición HTTP POST** con los datos del e-mail y la estación seleccionada, dirigidos al archivo `verificar.php`.

## d. Respuesta del servidor a la POST

El servidor procesó la información y devolvió una respuesta HTML dinámica con los resultados de la encuesta o, en caso de e-mail repetido, el mensaje de restricción. Esta respuesta también fue registrada como un paquete HTTP en Wireshark

Las capturas de estos cuatro paquetes fueron guardadas y se adjuntan al presente informe como evidencia del correcto funcionamiento de la aplicación en cuanto al intercambio de información entre cliente y servidor.



Integrantes: Martina Nahman y Emiliano Germani

ArchivoEdiciónVisualizaciónIrCapturaAnalizarEstadísticasTelefoníaWirelessHerramientasAyuda

http

No.	Time	Source	Destination	Protocol	Length	Info
34	19.973328309	10.65.3.225	10.65.3.40	HTTP	672	POST /verificar.php HTTP/1.1 (application/x-www-form-urlencoded)
36	19.984689783	10.65.3.40	10.65.3.225	HTTP	963	HTTP/1.1 200 OK (text/html)
37	20.085606512	10.65.3.225	10.65.3.40	HTTP	487	GET /style.css HTTP/1.1
39	20.086619246	10.65.3.40	10.65.3.225	HTTP	1083	HTTP/1.1 200 OK (text/css)
44	24.075751683	10.65.3.225	10.65.3.40	HTTP	506	GET /index.html? HTTP/1.1
46	24.077159142	10.65.3.40	10.65.3.225	HTTP	87	HTTP/1.1 200 OK (text/html)
81	34.342235336	10.65.3.225	10.65.3.40	HTTP	678	POST /verificar.php HTTP/1.1 (application/x-www-form-urlencoded)
83	34.346629584	10.65.3.40	10.65.3.225	HTTP	965	HTTP/1.1 200 OK (text/html)

Frame 37: 487 bytes on wire (3896 bits), 487 bytes captured (3896 bits) on interface wlp3s0, id 0

Ethernet II, Src: CloudNetwork\_b7:46:33 (38:03:c8:b7:46:33), Dst: LiteonTechno\_64:84:fd (3c:a0:07:64)

Internet Protocol Version 4, Src: 10.65.3.225, Dst: 10.65.3.40

Transmission Control Protocol, Src Port: 54655, Dst Port: 80, Seq: 619, Ack: 910, Len: 433

Hypertext Transfer Protocol

3c a0 67 64 84 fd 30 03 c8 b7 46 33 08 00 45 00 <gd-0> F3 E

01 d9 5f 98 40 00 80 96 7d fc 0a 41 03 e1 0a 41 -[0-9] A A

03 28 d5 7f 00 50 a4 4b 53 fc c4 e5 f6 4b 50 18 (-P K S KP

00 fc 9b ad 00 00 47 45 54 20 2f 73 74 79 6c 05 .....GE T /style

2e 63 73 73 20 48 54 54 50 2f 31 2e 31 0d 0a 48 .css HTTP/1.1 H

6f 73 74 3a 20 31 30 2e 36 35 2e 33 2e 34 30 0d ost: 10.65.3.40

0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 05 .Connect ion: kee

70 2d 61 6c 69 76 65 0d 0a 55 73 65 72 2d 41 07 p-alive- User-Ag

65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 ent: Mozilla/5.0

20 28 57 69 6e 64 6f 77 73 29 4e 54 20 31 30 2e (Window s NT 10.

30 3b 20 57 69 6e 36 34 3b 20 78 36 34 29 20 41 0; Win64 ; x64) A

70 70 6c 65 57 65 62 4b 69 74 2f 35 33 37 2e 33 ppleWebK it/537.3

36 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 6 (KHTML, Like G

65 63 6b 6f 29 20 43 68 72 6f 6d 65 2f 31 33 36 ecko) Chrome/136

2e 30 2e 30 2e 30 20 53 61 66 61 72 69 2f 35 33 .0.0.0 Safari/53

37 2e 33 30 0d 0a 49 66 2d 4d 6f 64 69 66 69 65 7.36.1f -Modifie

64 2d 53 69 6e 63 65 3a 20 57 65 64 2c 20 32 38 d-Since: Wed, 28

20 4d 61 79 20 32 30 32 35 29 31 37 3a 34 30 3a May 2025 17:40:

30 37 20 47 4d 54 0d 0a 49 66 2d 4e 6f 6e 65 2d 07 GMT - If-None-

4d 61 74 63 68 3a 20 22 36 37 65 2d 36 33 36 33 Match: "67e-6363

35 61 62 66 65 36 30 61 64 2d 67 7a 69 70 22 0d Sabfe60a d-gzip"

0a 41 63 63 65 70 74 3a 20 74 65 78 74 2f 63 73 -Accept: text/cs

73 2c 2a 2f 2a 3b 71 3d 30 2e 31 0d 0a 53 65 63 s,\*/";q=0.1- Sec

2d 47 50 43 3a 20 31 0d 0a 41 63 63 65 70 74 2d -GPC: 1 -Accept-

4c 61 6e 67 75 61 6f 65 3a 20 65 73 2d 34 31 39 Language : es-419

2c 65 73 3b 71 3d 30 2e 35 0d 0a 52 65 66 65 72 ,es;q=0.5- Refer

Hypertext Transfer Protocol: Protocol

Paquetes: 1239 - Mostrado: 8 (0.6%)

Perfil: Default