

Instituto Politécnico Nacional

Escuela Superior de Cómputo

*Aplicaciones para comunicaciones de red*

*“Práctica 4 – Servidor HTTP”*

**Grupo:** 3CM15

**Integrantes:**

* Ramírez Olvera Guillermo.
* Sánchez Méndez Edmundo Josue.

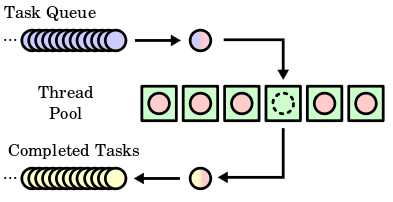
**Fecha de entrega:** 17 de mayo de 2021

**Profesor:** Moreno Cervantes Axel Ernesto



En la programación de computadoras, un grupo de subprocesos es un patrón de diseño de software para lograr la concurrencia de ejecución en un programa de computadora. A menudo también llamado modelo de trabajadores replicados o modelo de trabajadores, un grupo de subprocesos mantiene varios subprocesos esperando que el programa de supervisión asigne tareas para la ejecución concurrente. Al mantener un grupo de subprocesos, el modelo aumenta el rendimiento y evita la latencia en la ejecución debido a la frecuente creación y destrucción de subprocesos para tareas de corta duración. El número de subprocesos disponibles se ajusta a los recursos informáticos disponibles para el programa, como una cola de tareas paralelas después de la finalización de la ejecución.

El tamaño de un grupo de subprocesos es el número de subprocesos que se mantienen en reserva para ejecutar tareas. Suele ser un parámetro ajustable de la aplicación, ajustado para optimizar el rendimiento del programa. Decidir el tamaño óptimo del grupo de subprocesos es crucial para optimizar el rendimiento. Se ha sugerido la técnica de análisis de grupo de subprocesos (HTA) basada en la hipérbola para determinar el tamaño óptimo del grupo de subprocesos para el proceso de indexación basado en la nube en función de la carga de trabajo y el ancho de banda disponibles.



**Mime**.

Un tipo MIME es una etiqueta utilizada para identificar un tipo de datos. Se utiliza para que el software pueda saber cómo manejar los datos. Tiene el mismo propósito en Internet que las extensiones de archivo en Microsoft Windows.

Entonces, si un servidor dice "Esto es texto / html", el cliente puede ir "Ah, este es un documento HTML, puedo representar internamente", mientras que si el servidor dice "Esto es aplicación / pdf", el cliente puede ir "Ah", Necesito iniciar el complemento FoxIt PDF Reader que el usuario ha instalado y que se ha registrado como el controlador de aplicaciones/pdf.

Los encontrará más comúnmente en los encabezados de los mensajes HTTP (para describir el contenido con el que responde un servidor HTTP o el formato de los datos que se PUBLICAN en una solicitud) y en encabezados de correo electrónico (para describir el formato del mensaje y archivos adjuntos).

**Desarrollo**

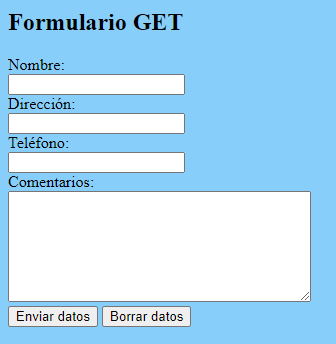
En esta práctica se hicieron peticiones como clientes a un servidor, que actúa como un apache, nos brinda un servicio con el método POST y GET.

Como primer lugar veamos que son el método GET:

Cuando un usuario rellena un formulario en una página web los datos hay que enviarlos de alguna manera. Vamos a considerar las dos formas de envío de datos posibles: usando el método POST

En el ejemplo anterior la acción que se ejecutará cuando el usuario pulse el botón “Enviar datos” (submit) será el envío de los datos a la url especificada usando el método get.

Veamos el aspecto de un formulario cualquiera para hacernos una idea general.

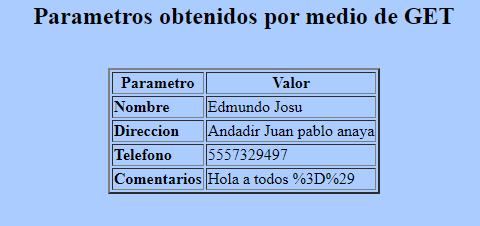


Este formulario consta de varios campos que al usuario se le solicitan como Nombre, Dirección, Teléfono y Comentarios.

La diferencia entre los métodos get y post radica en la forma de enviar los datos a la página cuando se pulsa el botón “Enviar”. Mientras que el método GET envía los datos usando la URL, el método POST los envía de forma que no podemos verlos (en un segundo plano u "ocultos" al usuario).

Un resultado usando el método GET, a modo de ejemplo, podría es el siguiente: http://localhost:7777/get?Nombre=Edmundo+Josu&Direccion=Andadir+Juan+pablo+anaya&Telefono=5557329497&Comentarios=Hola+a+todos+%3D%29

El símbolo ? indica dónde empiezan los parámetros que se reciben desde el formulario que ha enviado los datos a la página y la pagina nos devuelve el siguiente resultado:

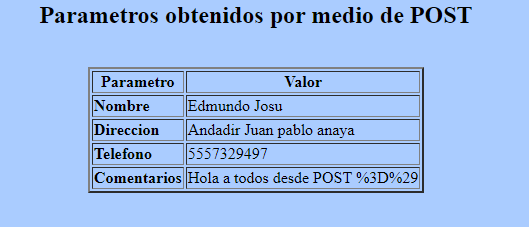


Ahora veamos el método POST:

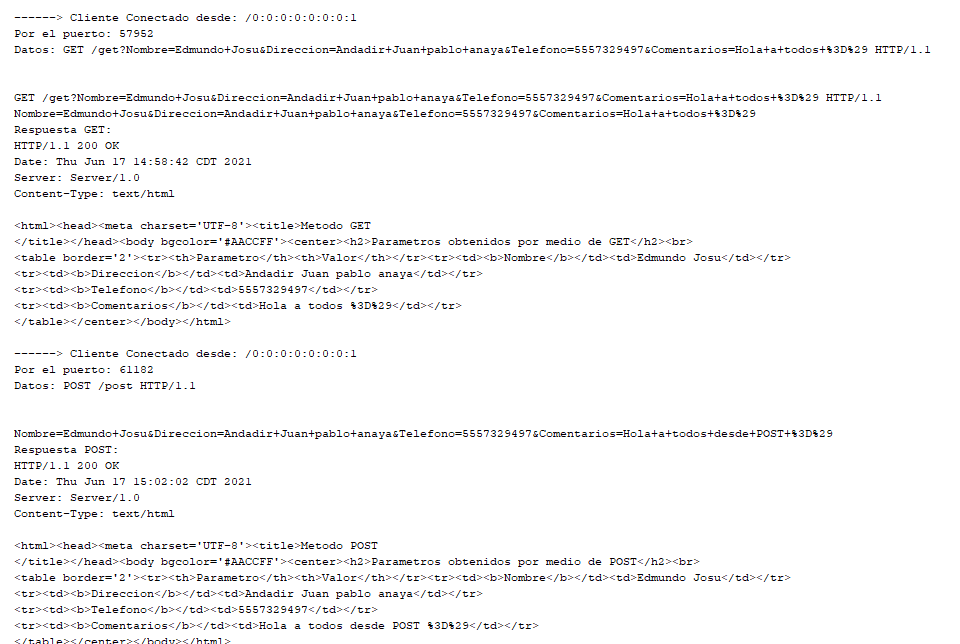
Hemos visto el resultado de un envío por el método GET. En el caso de un envío de datos usando el método POST, aunque estos datos también serán enviados (de una forma que podemos denominar “oculta”), no los podremos ver en la URL. Para poder recuperar los valores de los campos en el caso de un envío con el método POST necesitaríamos otras herramientas (por ejemplo, valernos del lenguaje PHP para recuperar el valor de esos campos).

El resultado final con ambos métodos podemos decir que es el mismo: la información se transmite de un lado a otro. La diferencia radica en que con el método GET podemos ver directamente los parámetros pasados ya que están dentro de la URL mientras que con el método POST los parámetros quedan ocultos y para rescatarlos hay que usar otras herramientas.

Un ejemplo de uso del método post sería este: <form action=" http://localhost:7777/post" method ="post">

A continuación, se muestran los parámetros obtenidos por el método POST: 

Por último, se muestran las peticiones de los clientes, estos se pueden observar cuando se corre el servidor, la siguiente captura muestra como cada cliente solicita consultas por método POST o GET:



Así mismo el servidor proporciona recursos disponibles como lo son imágenes, música, etc, que el usuario puede visualizar o acceder a ellos.



Y si el servidor borra uno de estos archivos y un cliente quiere acceder a él se muestra:



En esta práctica se puede observar de manera perfecta el uso del pool de hilos, ya que este permite varias conexiones sin que este se vicie, además, se pudo observar el uso de los sockets normales, en esta ocasión para levantar un servidor, además, pudimos crear un servidor HTTP basado en el propio protocolo y hacer la réplica del servidor con los estándares mime para poder hacer la visualización de cualquier elemento que tengamos en nuestra computadora, como, por ejemplo: Un archivo con extensión mp3, o con extensión PDF.