



INSTITUTO POLITÉCNICO NACIONAL.

ESCUELA SUPERIOR DE CÓMPUTO.

**APLICACIONES PARA COMUNICACIONES
EN RED.**

PRÁCTICA 5.

**PROFESOR: AXEL ERNESTO MORENO
CERVANTES.**

GRUPO 6CM1.

**ALUMNA: SÁNCHEZ VALERIANO
ALEXANDRA.**

FECHA: 05/01/2026

INTRODUCCIÓN.

La descarga de información masiva y la gestión eficiente de conexiones de red son pilares fundamentales en la computación distribuida. Herramientas como Wget son estándares en sistemas UNIX para la recuperación de contenido web. Sin embargo, la implementación clásica basada en Sockets Bloqueantes (java.io) presenta limitaciones de escalabilidad, ya que requiere un hilo por conexión, consumiendo excesivos recursos del sistema operativo.

Esta práctica se enfoca en la implementación de un cliente Wget en Java utilizando la arquitectura NIO (New I/O) y Sockets No Bloqueantes. A diferencia del modelo tradicional, NIO permite gestionar múltiples canales de comunicación ([SocketChannel1](#)) mediante un único hilo utilizando un Selector. Esto permite monitorear eventos de red (conexión, lectura, escritura) de manera asíncrona.

El objetivo principal es desarrollar un cliente capaz de conectarse a un servidor HTTP (Apache simulado), descargar archivos de manera recursiva, interpretar la estructura de directorios y gestionar el almacenamiento local, todo ello bajo un paradigma no bloqueante y respetando el protocolo HTTP/1.1.

DESARROLLO.

1. ANÁLISIS DEL SERVIDOR (SERVERWGET.JAVA).

El servidor desarrollado actúa como el contraparte necesario para probar el cliente. Su funcionamiento inicia con la creación de un **ServerSocket** atado al puerto 8000, el cual entra en un ciclo infinito esperando conexiones entrantes. A diferencia de servidores comerciales complejos, este diseño simplificado instancia un nuevo hilo por cada cliente aceptado, delegando el procesamiento a la función **manejarCliente**.

Dentro de la lógica de manejo del cliente, el servidor utiliza flujos de entrada amortiguados (**BufferedInputStream**) para leer la petición HTTP cruda. El primer paso crucial es el análisis de la primera línea de la petición, donde se extrae el método (usualmente GET) y la ruta del recurso solicitado. El código incluye una normalización de la ruta para decodificar caracteres especiales URL y redirigir la raíz ("/") automáticamente al archivo **index.html**.

Una parte esencial del servidor es la identificación correcta de los tipos MIME. El sistema verifica la extensión del archivo solicitado en el sistema de archivos local y asigna la cabecera **Content-Type** correspondiente, soportando HTML, texto plano, imágenes JPEG o PNG y documentos PDF. Si el archivo existe, se leen todos sus bytes y se construye una respuesta HTTP/1.1 200 OK que incluye la fecha, el servidor, la longitud del contenido y el tipo de archivo, seguido de los bytes del archivo. En caso contrario, se genera dinámicamente una respuesta 404 Not Found. Esta implementación asegura que el cliente reciba las cabeceras necesarias para procesar correctamente los datos binarios o de texto.

2. ARQUITECTURA DEL CLIENTE (CLIENTEWGET.JAVA).

La clase **ClienteWget** es el núcleo de la práctica y combina una interfaz gráfica basada en Swing con una lógica de red avanzada. La interfaz se divide en un panel superior para la configuración de la URL y directorio de destino, y un panel dividido que muestra la tabla de progreso de descargas y un área de pestañas para visualizar los logs de comunicación o la vista previa del sitio descargado.

2.1 IMPLEMENTACIÓN DE JAVA NIO.

El método **procesarArchivoNIO** contiene la lógica más compleja del sistema. En lugar de usar **Socket**, se utiliza **SocketChannel.open()**, el cual se configura inmediatamente en modo no bloqueante mediante **configureBlocking(false)**. Este canal se registra en un objeto **Selector** con la operación **OP_CONNECT**. El flujo de control depende de un ciclo **while** que interroga al selector. Si el canal está listo para conectarse, se finaliza el proceso

de conexión TCP y se envía manualmente la petición HTTP GET construida con cadenas de texto, registrando luego el canal para operaciones de lectura (**OP_READ**).

Cuando el selector detecta datos disponibles para lectura, se utiliza un **ByteBuffer** para extraer los datos del canal. Aquí reside el desafío principal de NIO y HTTP: los datos llegan fragmentados y mezclados. El código implementa una lógica para acumular los bytes recibidos en un **ByteArrayOutputStream** hasta detectar la secuencia de bytes **\r\n\r\n\r\n**, que marca el fin de las cabeceras HTTP y el inicio del cuerpo del archivo. Una vez separadas las cabeceras, el resto de los bytes se escriben directamente en un **FileOutputStream** para guardar el archivo en el disco local, asegurando la integridad de archivos binarios como imágenes.

2.2 LÓGICA RECURSIVA Y PARSING.

Para lograr el comportamiento de "araña web", el cliente mantiene una estructura de datos tipo cola (**Queue**) llamada **colaDescargas** y un conjunto (**Set**) de URLs visitadas. Cuando se descarga un archivo con extensión HTML o PHP, se invoca al método **analizarHtml**. Este método lee el contenido del archivo descargado y utiliza Expresiones Regulares (**Pattern** y **Matcher**) para buscar patrones de etiquetas **src="..."** y **href="..."**.

El algoritmo extrae los enlaces encontrados dentro del documento, valida que no sean enlaces externos absolutos (que comiencen con http) o anclas, y construye la ruta relativa completa. Si el recurso no ha sido visitado previamente, se añade a la cola de descargas y se actualiza la tabla de la interfaz gráfica. Este proceso se repite cíclicamente en un hilo secundario hasta que la cola de descargas queda vacía, logrando así descargar todo el árbol de dependencias del sitio web (imágenes, hojas de estilo, scripts) de manera automática.

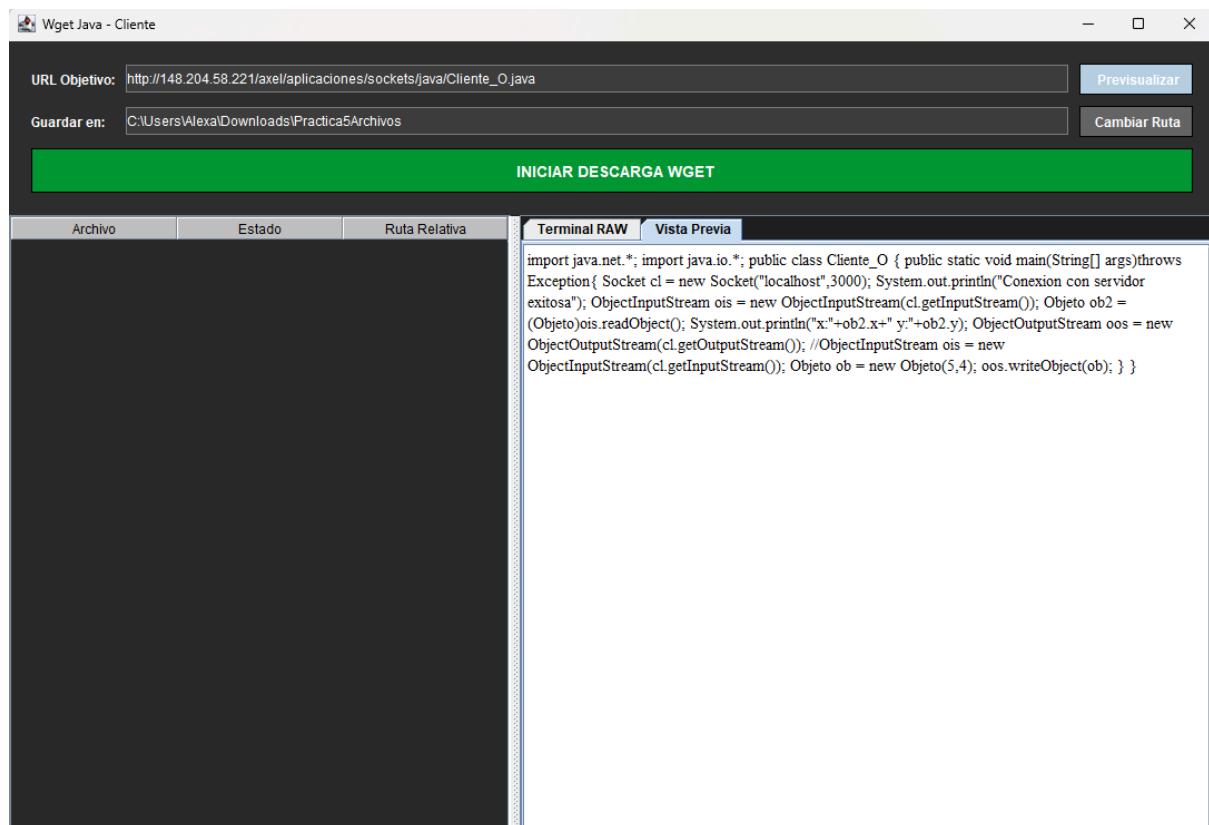
PRUEBAS.

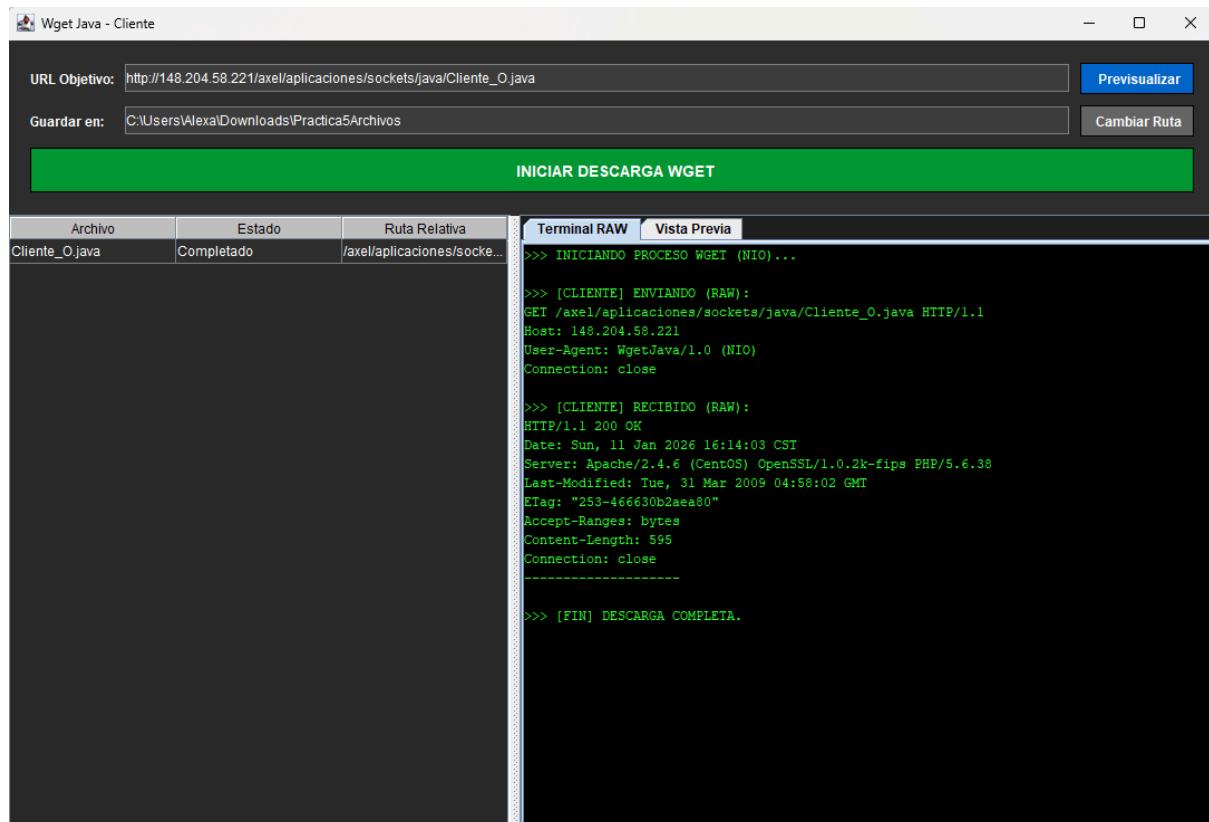
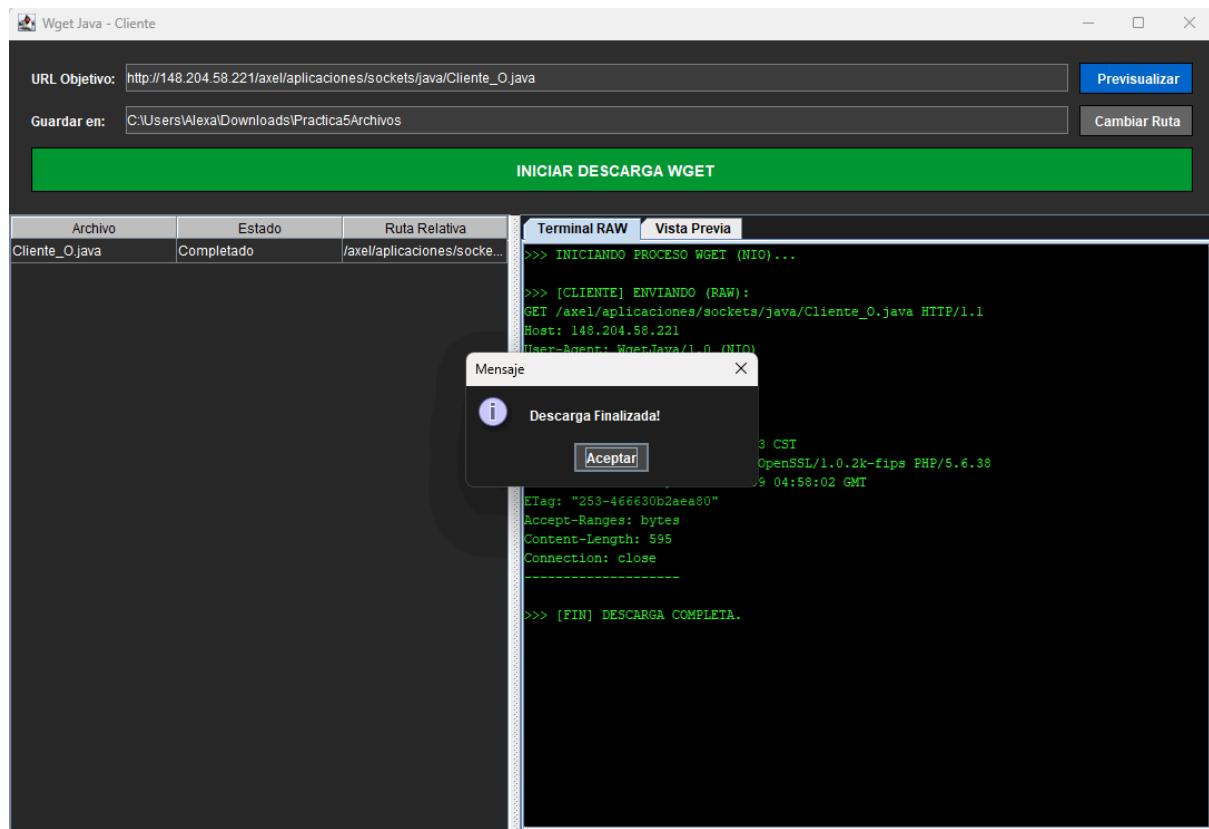
run:

=====

SERVER APACHE - PUERTO 8000

=====





Wget Java - Cliente

URL Objetivo: http://148.204.58.221/axel/aplicaciones/sockets/java/Cliente_O.java
Guardar en: C:\Users\AlexaDownloads\Practica5Archivos

INICIAR DESCARGA WGET

Archivo	Estado	Ruta Relativa
Cliente_O.java	Completo	/axel/aplicaciones/socke...

Terminal RAW Vista Previa

```
import java.net.*;
import java.io.*;

public class Cliente_O {

public static void main(String[] args) throws Exception{
    Socket cl = new Socket("localhost",3000);
    System.out.println("Conexion con servidor exitosa");
    ObjectInputStream ois = new ObjectInputStream(cl.getInputStream());
    Objeto ob2 = (Objeto)ois.readObject();
    System.out.println("x:"+ob2.x+" y:"+ob2.y);

    ObjectOutputStream oos = new ObjectOutputStream(cl.getOutputStream());
    //ObjectInputStream ois = new ObjectInputStream(cl.getInputStream());
    Objeto ob = new Objeto(5,4);
    oos.writeObject(ob);
}

}
```

C Descargas > Practica5Archivos > axel > aplicaciones > sockets > java

Ordenar Ver ...

Nombre	Fecha de modificación	Tipo	Tamaño
Cliente_O	11/01/2026 04:14 p. m.	Archivo de origen ...	1 KB

Wget Java - Cliente

URL Objetivo: <http://148.204.58.221/axel/aplicaciones/sockets/java/>

Guardar en: C:\Users\Alexa\Downloads\Practica5Archivos

INICIAR DESCARGA WGET

Archivo	Estado	Ruta Relativa
Cliente_O.java	Completado	/axel/aplicaciones/sock...

Index of /axel/aplicaciones/sockets/java

Name	Last modified	Size	Description
Parent Directory		-	
Cliente_O.java	2009-03-30 22:58	595	
Cliente_ftp.java	2009-03-21 18:29	800	
DatagramPacket.txt	2011-05-04 09:59	4.2K	
DatagramaECO.zip	2023-03-29 09:53	27K	
DatagramaHM.zip	2020-09-29 10:32	19K	
DatagramaObjeto.zip	2023-03-29 07:34	24K	
EchoClient.java	2009-04-14 06:33	2.6K	
EchoServer.java	2009-04-14 06:32	1.4K	
FTP.zip	2012-08-15 06:27	124K	
FlujoArchivo.zip	2024-03-11 07:48	26K	
FlujoArchivo_modific...	2021-08-27 09:34	15M	
FlujoEco.zip	2021-09-23 08:23	27K	
Flujo_Objeto.zip	2023-03-07 08:08	26K	

Wget Java - Cliente

URL Objetivo: <http://148.204.58.221/axel/aplicaciones/sockets/java/>

Guardar en: C:\Users\Alexa\Downloads\Practica5Archivos

INICIAR DESCARGA WGET

Archivo	Estado	Ruta Relativa
java	Completado	/axel/aplicaciones/sock...
blank.gif	Completado	/axel/aplicaciones/sock...
back.gif	Completado	/axel/aplicaciones/sock...
sockets	Completado	/axel/aplicaciones/sock...
unknown.gif	Completado	/axel/aplicaciones/sock...
Cliente_O.java	Completado	/axel/aplicaciones/sock...
Cliente_ftp.java	Completado	/axel/aplicaciones/sock...
text.gif	Completado	/axel/aplicaciones/sock...
DatagramPacket.txt	Completado	/axel/aplicaciones/sock...
compressed.gif	Completado	/axel/aplicaciones/sock...
DatagramaECO.zip	Completado	/axel/aplicaciones/sock...
DatagramaHM.zip	Completado	/axel/aplicaciones/sock...
DatagramaObjeto.zip	Completado	/axel/aplicaciones/sock...
EchoClient.java	Completado	/axel/aplicaciones/sock...
EchoServer.java	Completado	/axel/aplicaciones/sock...
FTP.zip	Completado	/axel/aplicaciones/sock...
FlujoArchivo.zip	Completado	/axel/aplicaciones/sock...
FlujoArchivo_modific...	Completado	/axel/aplicaciones/sock...
FlujoEco.zip	Completado	/axel/aplicaciones/sock...
Flujo_Objeto.zip	Completado	/axel/aplicaciones/sock...
GUI.zip	Completado	/axel/aplicaciones/sock...
HolaMundo.zip	Completado	/axel/aplicaciones/sock...
HolaMundo999.zip	Completado	/axel/aplicaciones/sock...
Multicast.zip	Completado	/axel/aplicaciones/sock...

Terminal RAW

```
User-Agent: WgetJava/1.0 (NIO)
Connection: close

>>> [CLIENTE] RECIBIDO (RAW):
HTTP/1.1 404 Not Found
Date: Sun, 11 Jan 2026 16:16:56 CST
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.6.38
Content-Length: 258
Connection: close
Content-Type: text/html; charset=iso-8859-1
-----
>>>

>>> [CLIENTE] ENVIANDO (RAW):
GET /axel/aplicaciones/sockets/java/canales/canales.zip HTTP/1.1
Host: 148.204.58.221
User-Agent: WgetJava/1.0 (NIO)
Connection: close

>>> [CLIENTE] RECIBIDO (RAW):
HTTP/1.1 200 OK
Date: Sun, 11 Jan 2026 16:16:56 CST
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.6.38
Last-Modified: Thu, 09 Oct 2025 22:47:08 GMT
ETag: "544e7-640c194c3471b"
Accept-Ranges: bytes
Content-Length: 345319
Connection: close
Content-Type: application/zip
-----
```

Wget Java - Cliente

URL Objetivo: Previsualizar

Guardar en: Cambiar Ruta

INICIAR DESCARGA WGET

Archivo	Estado	Ruta Relativa
CEcoFlujo.java	Completado	/axel/aplicaciones/sock...
SEcoFlujo.java	Completado	/axel/aplicaciones/sock...
blank.gif	Completado	/axel/aplicaciones/sock...
back.gif	Completado	/axel/aplicaciones/sock...
recurso	Completado	/axel/aplicaciones/sock...
unknown.gif	Completado	/axel/aplicaciones/sock...
Cliente_O.java	Completado	/axel/aplicaciones/sock...
text.gif	Completado	/axel/aplicaciones/sock...
Objeto.java	Completado	/axel/aplicaciones/sock...
Servidor_O.java	Completado	/axel/aplicaciones/sock...
blank.gif	Completado	/axel/aplicaciones/sock...
back.gif	Completado	/axel/aplicaciones/sock...
udp	Completado	/axel/aplicaciones/sock...
unknown.gif	Completado	/axel/aplicaciones/sock...
Cliente_O_UDP.java	Completado	/axel/aplicaciones/sock...
Objeto_U.java	Completado	/axel/aplicaciones/sock...
Servidor_O_UDP.java	Completado	/axel/aplicaciones/sock...
blank.gif	Completado	/axel/aplicaciones/sock...
back.gif	Completado	/axel/aplicaciones/sock...
udp	Completado	/axel/aplicaciones/sock...
unknown.gif	Completado	/axel/aplicaciones/sock...
Cliente_O_UDP.java	Completado	/axel/aplicaciones/sock...
Objeto_U.java	Completado	/axel/aplicaciones/sock...
Servidor_O_UDP.java	Completado	/axel/aplicaciones/sock...

Terminal RAW Vista Previa

```
>>> [CLIENTE] RECIBIDO (RAW):
HTTP/1.1 200 OK
Date: Sun, 11 Jan 2026 16:17:10 CST
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.6.38
Last-Modified: Mon, 17 Oct 2011 11:15:10 GMT
ETag: "199-4af7cb8518b80"
Accept-Ranges: bytes
Content-Length: 409
Connection: close
-----
>>> [CLIENTE] ENVIANDO (RAW):
GET /axel/aplicaciones/sockets/java/udp/serializacion/Servidor_O_UDP.java HTTP/1.1
Host: 148.204.58.221
User-Agent: WgetJava/1.0 (NIO)
Connection: close

>>> [CLIENTE] RECIBIDO (RAW):
HTTP/1.1 200 OK
Date: Sun, 11 Jan 2026 16:17:10 CST
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.6.38
Last-Modified: Mon, 17 Oct 2011 11:48:46 GMT
ETag: "515-4af7d307b4380"
Accept-Ranges: bytes
Content-Length: 1301
Connection: close
-----
>>> [FIN] DESCARGA COMPLETA.
```

Wget Java - Cliente

URL Objetivo: <http://148.204.58.221/axel/aplicaciones/sockets/java/>

Guardar en: C:\Users\AlexaDownloads\Practica5Archivos

INICIAR DESCARGA WGET

Archivo	Estado	Ruta Relativa
CEcoFlujo.java	Completado	/axel/aplicaciones/sock...
SEcoFlujo.java	Completado	/axel/aplicaciones/sock...
blank.gif	Completado	/axel/aplicaciones/sock...
back.gif	Completado	/axel/aplicaciones/sock...
recurso	Completado	/axel/aplicaciones/sock...
unknown.gif	Completado	/axel/aplicaciones/sock...
Cliente_O.java	Completado	/axel/aplicaciones/sock...
text.gif	Completado	/axel/aplicaciones/sock...
Objeto.java	Completado	/axel/aplicaciones/sock...
Servidor_O.java	Completado	/axel/aplicaciones/sock...
blank.gif	Completado	/axel/aplicaciones/sock...
back.gif	Completado	/axel/aplicaciones/sock...
udp	Completado	/axel/aplicaciones/sock...
unknown.gif	Completado	/axel/aplicaciones/sock...
Cliente_O_UDP.java	Completado	/axel/aplicaciones/sock...
Objeto_U.java	Completado	/axel/aplicaciones/sock...
Servidor_O_UDP.java	Completado	/axel/aplicaciones/sock...
blank.gif	Completado	/axel/aplicaciones/sock...
back.gif	Completado	/axel/aplicaciones/sock...
udp	Completado	/axel/aplicaciones/sock...
unknown.gif	Completado	/axel/aplicaciones/sock...
Cliente_O_UDP.java	Completado	/axel/aplicaciones/sock...
Objeto_U.java	Completado	/axel/aplicaciones/sock...
Servidor_O_UDP.java	Completado	/axel/aplicaciones/sock...

Terminal RAW Vista Previa

```
import java.io.Serializable;

public class Objeto_U implements Serializable {
    String nombre;
    int edad;
    transient String pwd;

    public Objeto_U(String nombre, int edad, String pwd){
        this.nombre = nombre;
        this.edad = edad;
        this.pwd = pwd;
    }

    public String getNombre(){return this.nombre;}
    public int getEdad(){return this.edad;}
    public String getPwd(){return this.pwd;}
}
```

C Descargas > Practica5Archivos > axel > aplicaciones > sockets > java

Ordenar Ver ...

Nombre	Fecha de modificación	Tipo	Tamaño
▼ hoy			
 captura_HTTP	11/01/2026 04:17 p. m.	Documento Adob...	40 KB
 datagrama	11/01/2026 04:17 p. m.	Documento de tex...	4 KB
 multicastC-JAVA	11/01/2026 04:17 p. m.	Carpeta de archiv...	3 KB
 Othello2	11/01/2026 04:17 p. m.	Carpeta comprimi...	51 KB
 PacketReceiveDemo	11/01/2026 04:17 p. m.	Archivo de origen ...	2 KB
 PacketSendDemo	11/01/2026 04:17 p. m.	Archivo de origen ...	2 KB
 Server_tmp	11/01/2026 04:17 p. m.	Carpeta comprimi...	55 KB
 Server2	11/01/2026 04:17 p. m.	Carpeta comprimi...	1,373 KB
 Servidor_ftp	11/01/2026 04:17 p. m.	Archivo de origen ...	2 KB
 Servidor_O	11/01/2026 04:17 p. m.	Archivo de origen ...	1 KB
 ServidorWeb3	11/01/2026 04:17 p. m.	Archivo de origen ...	5 KB
 Tarjeta	11/01/2026 04:17 p. m.	Archivo de origen ...	2 KB
 TiposDatos	11/01/2026 04:17 p. m.	Carpeta comprimi...	11 KB
 GUI	11/01/2026 04:17 p. m.	Carpeta comprimi...	1,615 KB
 HolaMundo	11/01/2026 04:17 p. m.	Carpeta comprimi...	31 KB
 HolaMundo999	11/01/2026 04:17 p. m.	Carpeta comprimi...	11 KB
 Multicast	11/01/2026 04:17 p. m.	Carpeta comprimi...	229 KB
 MulticastClient	11/01/2026 04:17 p. m.	Archivo de origen ...	2 KB
 MulticastClient2	11/01/2026 04:17 p. m.	Archivo de origen ...	2 KB
 MulticastServer2	11/01/2026 04:17 p. m.	Archivo de origen ...	2 KB
 Mutex	11/01/2026 04:17 p. m.	Archivo de origen ...	2 KB
 Objeto	11/01/2026 04:17 p. m.	Archivo de origen ...	1 KB
 Othello	11/01/2026 04:17 p. m.	Carpeta comprimi...	31 KB
 Flujo_Objeto	11/01/2026 04:17 p. m.	Carpeta comprimi...	26 KB
 FlujoArchivo_modificado	11/01/2026 04:17 p. m.	Carpeta comprimi...	15,741 KB
 FlujoEco	11/01/2026 04:17 p. m.	Carpeta comprimi...	28 KB
 FlujoArchivo	11/01/2026 04:16 p. m.	Carpeta comprimi...	26 KB
 FTP	11/01/2026 04:16 p. m.	Carpeta comprimi...	125 KB
 Cliente_ftp	11/01/2026 04:16 p. m.	Archivo de origen ...	1 KB

CONCLUSIONES.

La realización de esta práctica me permitió comprender a profundidad las diferencias arquitectónicas y de rendimiento entre el modelo de entrada/salida bloqueante tradicional y la E/S no bloqueante (NIO) de Java. Al implementar el cliente Wget, se hizo evidente que, aunque la programación con **Channels** y **Selectors** introduce una mayor complejidad lógica en comparación con los **Sockets** convencionales, ofrece una ventaja significativa en la gestión de recursos. Esto se comprobó al observar cómo un solo hilo de ejecución podía administrar la descarga simultánea de múltiples archivos (imágenes, scripts y HTML) sin la necesidad de instanciar un hilo pesado del sistema operativo por cada conexión, lo cual es vital para la escalabilidad de aplicaciones de red modernas.

Asimismo, la manipulación directa del protocolo HTTP a nivel de bytes reforzó mi entendimiento sobre la estructura de los paquetes de red. Fue indispensable implementar un análisis meticuloso de los buffers para detectar correctamente la secuencia de fin de cabecera (`\r\n\r\n\r\n`); sin este paso crítico, era imposible separar con precisión los metadatos de la respuesta del contenido binario real, lo que resultaba en archivos corruptos, especialmente al descargar imágenes o documentos PDF. Este proceso demostró que, más allá de utilizar librerías de alto nivel, un ingeniero debe ser capaz de procesar flujos de datos crudos para garantizar la integridad de la información.

Finalmente, la integración de una lógica recursiva para el "crawling" o rastreo de enlaces expuso la complejidad subyacente en los navegadores web actuales. La necesidad de analizar el código HTML en busca de etiquetas específicas, normalizar las rutas relativas y gestionar una cola de descargas asíncrona permitió simular con éxito el comportamiento de herramientas comerciales, validando la importancia de estructuras de datos eficientes en el desarrollo de software de comunicaciones.

REFERENCIAS BIBLIOGRÁFICAS.

- Fielding, R., et al. (1999). RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1. IETF.
- Oracle. (2024). Java SE Documentation: java.nio.channels.Selector & SocketChannel. Recuperado de <https://docs.oracle.com/en/java/javase/>
- Harold, E. R. (2013). Java Network Programming. O'Reilly Media.
- Mozilla Developer Network (MDN). (2025). HTTP Headers & Content-Type. Recuperado de <https://developer.mozilla.org/es/docs/Web/HTTP>