



**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

**Aplicaciones para Comunicaciones en Red**



## **Practica 4**

### **Sockets de datagrama**

Nombre Del Alumno: García Quiroz Gustavo Ivan

Grupo: 6CM4

Nombre de la Profesora: Sandra Ivette Bautista Rosales

## **Sockets de datagrama**

Esta práctica consistió en implementar un programa cliente-servidor utilizando sockets de datagrama (UDP) con la capacidad de enviar y recibir mensajes fragmentados en múltiples datagramas. El objetivo principal fue comprender el funcionamiento del protocolo UDP y cómo manejar la fragmentación de mensajes cuando estos exceden el tamaño máximo de un datagrama.

A diferencia del protocolo TCP, UDP no garantiza que los paquetes lleguen en orden ni que lleguen todos, por lo que fue necesario implementar un mecanismo para asegurar la integridad de los mensajes completos.

## **Herramientas utilizadas**

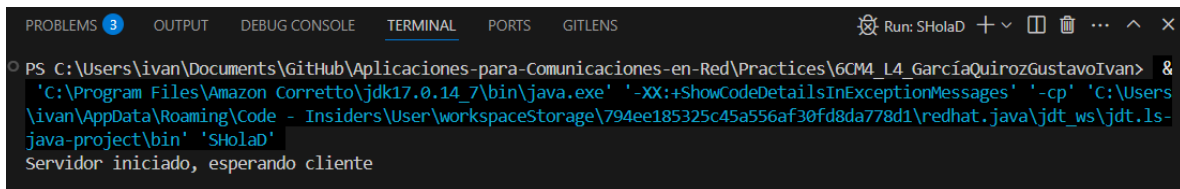
- **IDE:** Visual Studio Code
- **JDK:** Java Development Kit 17
- **Extensiones VS Code:**
  - Extension Pack for Java
  - Maven for Java
  - Project Manager for Java
  - Test Runner for Java

## **Pruebas y capturas de ejecución**

Para probar el funcionamiento del programa, se realizaron varias pruebas con mensajes de diferentes longitudes para verificar la fragmentación y reconstrucción de mensajes.

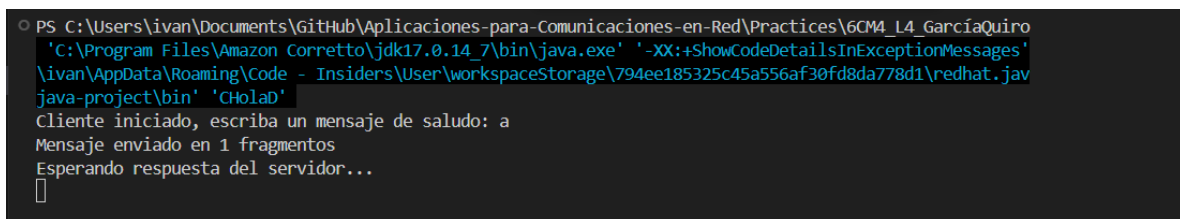
## **Mensajes cortos**

- **Captura 1: Ejecución del servidor**



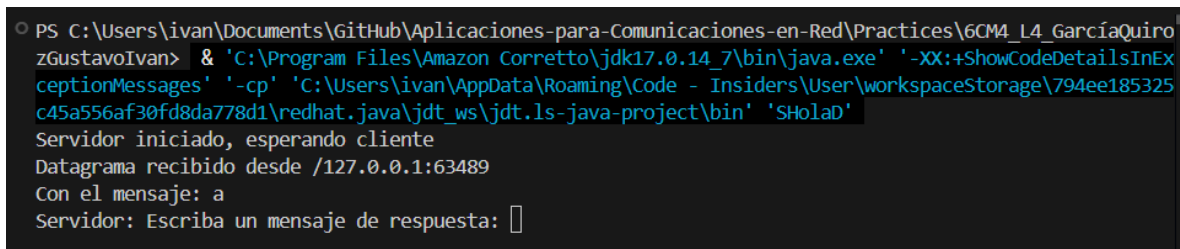
```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS Run: SHolaD + - - - ^ x
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuirozGustavoIvan> & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code - Insiders\User\workspaceStorage\794ee185325c45a556af30fd8da778d1\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'SHolaD'
Servidor iniciado, esperando cliente
```

- **Captura 2: Ejecución del cliente enviando mensaje corto**



```
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuirozGustavoIvan> & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code - Insiders\User\workspaceStorage\794ee185325c45a556af30fd8da778d1\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'CHolaD'
Cliente iniciado, escriba un mensaje de saludo: a
Mensaje enviado en 1 fragmentos
Esperando respuesta del servidor...
█
```

- **Captura 3: Servidor recibiendo mensaje corto**



```
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuirozGustavoIvan> & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code - Insiders\User\workspaceStorage\794ee185325c45a556af30fd8da778d1\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'SHolaD'
Servidor iniciado, esperando cliente
Datagrama recibido desde /127.0.0.1:63489
Con el mensaje: a
Servidor: Escriba un mensaje de respuesta: █
```

En las capturas anteriores se puede observar que para mensajes cortos, el sistema envía pocos fragmentos (4 datagramas).

## Mensajes grandes (múltiples datagramas)

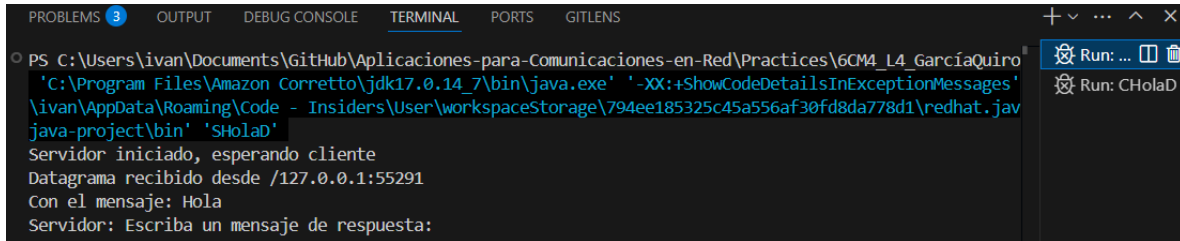
Para probar la fragmentación de mensajes grandes, se enviaron mensajes que exceden significativamente el tamaño del buffer (BUFFER\_SIZE = 8).

- **Captura 4: Cliente enviando mensaje grande**



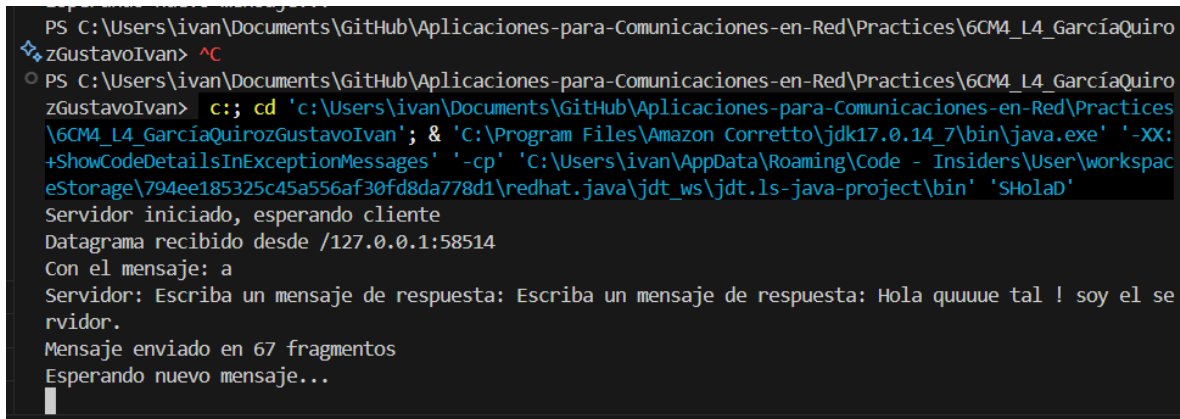
```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS Run: SHolaD + - - - ^ x Run: CHolaD
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuirozGustavoIvan> & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code - Insiders\User\workspaceStorage\794ee185325c45a556af30fd8da778d1\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'CHolaD'
Cliente iniciado, escriba un mensaje de saludo: Hola
Mensaje enviado en 4 fragmentos
Esperando respuesta del servidor...
█
```

- **Captura 6: Servidor recibiendo y procesando múltiples fragmentos**



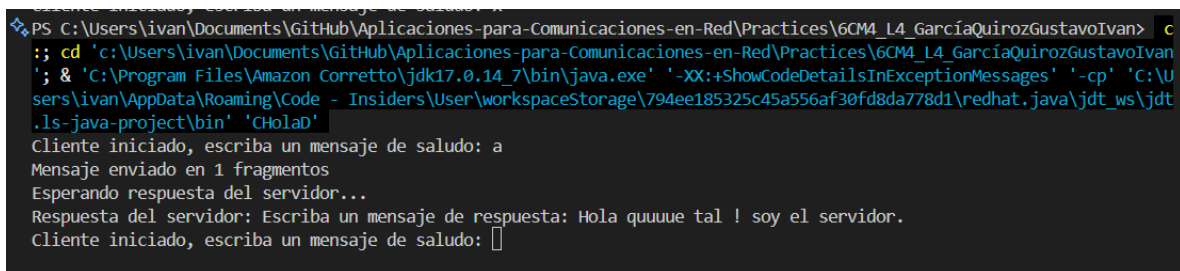
```
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuiroz> 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' \
ivan\AppData\Roaming\Code - Insiders\User\workspaceStorage\794ee185325c45a556af30fd8da778d1\redhat.java \
java-project\bin' 'SHolaD'
Servidor iniciado, esperando cliente
Datagrama recibido desde /127.0.0.1:55291
Con el mensaje: Hola
Servidor: Escriba un mensaje de respuesta:
```

- **Captura 7: Servidor enviando respuesta grande fragmentada**



```
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuiroz> zGustavoIvan> ^C
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuiroz> zGustavoIvan> c;; cd 'c:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\
6CM4_L4_GarcíaQuirozGustavoIvan'; & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code - Insiders\User\workspaceStorage\794ee185325c45a556af30fd8da778d1\redhat.java\jdt_ws\jdt
.ls-java-project\bin' 'SHolaD'
Servidor iniciado, esperando cliente
Datagrama recibido desde /127.0.0.1:58514
Con el mensaje: a
Servidor: Escriba un mensaje de respuesta: Escriba un mensaje de respuesta: Hola quuuue tal ! soy el servidor.
Mensaje enviado en 67 fragmentos
Esperando nuevo mensaje...
```

- **Captura 8: Cliente recibiendo y reconstruyendo respuesta fragmentada**



```
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuirozGustavoIvan> c
;; cd 'c:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Practices\6CM4_L4_GarcíaQuirozGustavoIvan
'; & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\U
sers\ivan\AppData\Roaming\Code - Insiders\User\workspaceStorage\794ee185325c45a556af30fd8da778d1\redhat.java\jdt_ws\jdt
.ls-java-project\bin' 'CHolaD'
Cliente iniciado, escriba un mensaje de saludo: a
Mensaje enviado en 1 fragmentos
Esperando respuesta del servidor...
Respuesta del servidor: Escriba un mensaje de respuesta: Hola quuuue tal ! soy el servidor.
Cliente iniciado, escriba un mensaje de saludo: 
```

Como se puede observar en las capturas, cuando se envía un mensaje grande (por ejemplo, de más de 67 caracteres), el sistema automáticamente lo divide en múltiples datagramas (fragmentos) y luego lo reconstruye en el destino.