

UPIITA---IPN---TELEMÁTICA Práctica 1

Sistema distribuidos

Introducción al Manejo de Sockets

NOMBRE ALUMNO: _____ GRUPO: _____

Objetivos:

Conocer como construir un Socket en Java , bajo la arquitectura CLIENTE-SERVIDOR y el enfoque de programación en red

Comprender el concepto de arquitectura

Comprender el concepto de transparencia en la comunicación

Conocer el manejo básico de Github

Resultados de aprendizaje

Al finalizar, el estudiante será capaz de:

- Implementar un servidor y un cliente por TCP usando sockets.
- Probar local y remotamente la comunicación (loopback y dos equipos).
- Documentar el protocolo de aplicación (formato de mensajes, delimitadores, encoding).
- Demostrar interoperabilidad Java↔C.

Requisitos:

- Un ambiente de red (alámbrica o inalámbrica)
- Tener instalado y configurado java , un IDE (e.g. Netbeans o Eclipse)
- Privilegios de administrador para configuración de la red, y de las directivas de seguridad del sistema operativo[permisos de root en Linux]].
- Descarga el código(los archivos java) desde GITHUB de la practica 1 en:

<https://github.com/migfel/practicass-SD-2022.git>

```
git clone https://github.com/migfel/practicass-SD-2022.git
```

```
cd practicas-SD-2022/eco
```

```
javac *.java
```

INSTRUCCIONES:

Las letras en color morado indican los que debes incluir en tu reporte.

I. Comunicación con Sockets Localmente

a) Abra una terminal o ventana de comandos, compile todos los archivos java que descargo de github y ejecute el programa *Servidor_Eco.java* indicando el puerto de su preferencia, en este caso usamos el 5000, **java Servidor_Eco 5000**

b) abra otra terminal y ejecute el programa *Cliente_de_Eco.java* indicando: el puerto, la IP o el nombre del servidor (el nombre que tiene tu computadora, o puedes usar localhost que es universal para referirse a tu propia computadora) . Ver ejemplo:

```
java Cliente_de_Eco 127.0.0.1 5000
```

c) Llama desde el cliente al programa *Servidor_Eco*, esto se hace al escribir un un mensaje de texto y dando enter.

Deberás observar en la consola/terminal, la respuesta que te envío el servidor, es decir, al mensaje que envío [que es precisamente el “eco” de lo que escribiste, (el mismo texto)].

El servidor Echo dice: “Mensaje que le enviaste”

De esta forma, estamos viendo paso de mensajes entre entidades de software

homogéneo (ambos programas están escritos en Java) el servidor usa la clase ServerSocket y el cliente usa la clase Socket

Responde en tu reporte ¿Que necesitamos para que pueda comunicarse el programa Servidor (codificado en java) con un cliente (codificado en C) y viceversa?

II---Comunicación con Sockets remotamente

Para este ejercicio se requiere que la ejecución se realice en una red WiFi o Ethernet, entre dos computadoras que pertenecen a la misma red o a diferentes redes, (si las redes son distintas, entonces debes poder indicar/gestionar la dirección IP de tu computadora como IP pública o en su defecto, puedes instalar un software de tipo MiddleWare para facilitar y lograr la comunicación entre tus programas (e.g. la herramienta Hamachi permite comunicar programas que están en diferentes redes).

a) Ejecute el programa EchoServer en una computadora A.

a) Ejecute el programa Cliente_de_Eco en la computadora B ;

Debe lograrse el mismo resultado de comunicación que en paso I inciso a)

Incluye en tu reporte las respuestas a estas preguntas:

- *¿cómo se llama esta característica/funcionalidad en un sistema distribuido?*
- *¿ Qué es lo que permite que esta característica ocurra?*

EJERCICIO DE TAREA

1.- Codifique dos programas usando sockets, en el enfoque cliente-servidor, que permita el intercambio de mensajes de texto .

- El programa Servidor (debe ser codificado en java)
- Mientras que el cliente (debe ser codificado en C)
- Funcionamiento: Cuando se conecten entre si, el cliente enviará una cadena de texto cualquiera, por ejemplo Hola y el Servidor debe responder con algún otro mensaje, por ejemplo, Hola que tal.

2.- Codifique dos programas usando sockets, en el enfoque cliente-servidor, que permita que se envíen números enteros entre si

El programa Servidor (debe ser codificado en C)

El programa cliente (debe ser codificado en Java)

- Funcionamiento: Cuando se conecten entre si, el cliente enviará un entero y el servidor lo incrementará en uno. Ejemplo, el cliente envía un 5 y el servidor contestará con un 6, el programa terminará cuando el cliente escriba un cero.

La prueba y revisión de este ejercicio consiste en:

Enviar el reporte documentando en el classroom, e incluye la liga de tu código en github
Explique o comente que retos encontró para resolver esto