



Instituto Politécnico Nacional Escuela Superior de Cómputo



Paradigmas de Programación

Actividad: Ejercicio 13

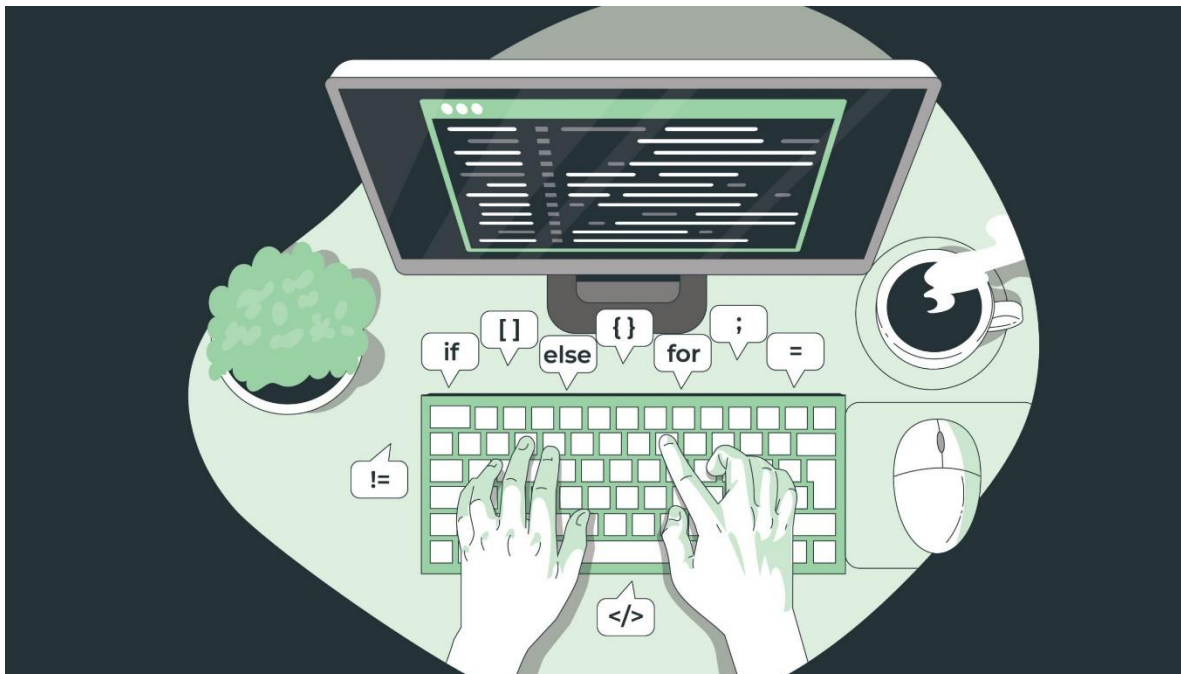
Profesor: García Floriano Andrés

Fecha: 20/06/2024

Alumno:

Pacheco Refugio Alan Ivan

Grupo: 3CV1



Resultados:

En el código ClienteJava, se agrega un bufferedreader esto con la finalidad de recibir y mostrar una respuesta enviada por un servidor y usando el out.println se envia un mensaje al servidor

```
try (Socket socket = new Socket(hostName, port)) {  
    PrintWriter out = new PrintWriter(socket.getOutputStream(), autoFlush:true);  
    //Se agrega un bufferedreader para recibir y mostrar la respuesta enviada por el servidor  
    BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
    out.println(x:"Hola, servidor!"); // Enviar un mensaje al servidor  
    System.out.println(x:"Mensaje enviado al servidor.");  
    String response = in.readLine(); // Leer la respuesta de parte del servidor  
    System.out.println("Respuesta del servidor: " + response);  
}
```

En el código de ServerJava, se agrega un printgritter para tener un mensaje del cliente y posteriormente dar una respuesta al cliente.

Después se lee los datos que el cliente ha enviado

```
// Abrir un stream para leer lo que el cliente está enviando  
BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));  
//Se agrega el print gritter, esto con la finalidad d poder enviar una resouesta al cliente, despues de un mensaje del cli  
PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), autoFlush:true);  
  
String inputLine = in.readLine(); // Leer los datos enviados por el cliente  
System.out.println("Mensaje del Cliente: " + inputLine);  
  
// Envio de una respuesta al cliente  
out.println(x:"Hola, Alan! Recibimos tu mensaje.");
```

Compilación:

Se compilan de esta manera para generar los “.class”generando los archivos “Client.class” y “Server.class”

```
ramación\Ejercicio de Laboratorio 13>java Client.java
```

```
de Programación>cd Ejercicio de Laboratorio 13
```

```
de Programación\Ejercicio de Laboratorio 13>java Server.java
```

Se ejecuta en una terminal aparte al ServerJava.class

```
C:\Users\ivano\OneDrive\Escritorio (3211)\4to Semestre>cd "Paradigmas de Programación"
C:\Users\ivano\OneDrive\Escritorio (3211)\4to Semestre\Paradigmas de Programación>cd "Ejercicio de Laboratorio 13"
C:\Users\ivano\OneDrive\Escritorio (3211)\4to Semestre\Paradigmas de Programación\Ejercicio de Laboratorio 13>java Server
Servidor iniciado, escuchando en el puerto: 5000
```

Posteriormente en una terminal aparte se ejecuta ClientJava.class, en donde se envía el mensaje al servidor y se obtiene una respuesta en esta caso Null

```
C:\Users\ivano\OneDrive\Escritorio (3211)\4to Semestre\Paradigmas de Programación\Ejercicio de Laboratorio 13>java Client
Mensaje enviado al servidor.
Respuesta del servidor: null
```

Una vez que se han compilado ambos códigos se da una envía un mensaje y se recibe una respuesta por parte del servidor

Cliente envía mensaje al servidor y recibe respuesta

```
C:\Users\ivano\OneDrive\Escritorio (3211)\4to Semestre\Paradigmas de Programación\Ejercicio de Laboratorio 13>java Client
Mensaje enviado al servidor.
Respuesta del servidor: null
```

¡Server inicia su rutina en el puerto 5000 en donde se conecta el cliente y se recibe un mensaje del cliente en este caso “¡Hola, Servidor!”

```
C:\Users\ivano\OneDrive\Escritorio (3211)\4to Semestre\Paradigmas de Programación\Ejercicio de Laboratorio 13>java Server
Servidor iniciado, escuchando en el puerto: 5000
Cliente conectado
Mensaje del Cliente: Hola, servidor!
```