Programación orientada a objetos con JAVA
Networking y Multitreading

Cliente Socket

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

PrintWriter out;
BufferedReader in;

try {
        Socket socket = new Socket("localhost",23);
        out = new PrintWriter(socket.getOutputStream(), true);
        in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        ....
        out.println(“enviando mensaje”);
        System.out.println(in.readLine();
        ...
} catch (Exception e) {
        e.printStackTrace();
}
```

Servidor Socket

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

ServerSocket servidor = null;
Socket socket = null;
PrintWriter out;
BufferedReader in;

try {
        servidor = new ServerSocket(23);
        socket = servidor.accept();
        out = new PrintWriter(socket.getOutputStream(), true);
        in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        ....
        out.println(“enviando mensaje”);
        System.out.println(in.readLine();
        ...

} catch (Exception e) {
        e.printStackTrace();
}
```

Hilos

Utilizando la clase Thread

```
public class Tarea extends Thread {

       public void run() {
            while (true) {
                  ...
                  this.sleep(1000);
                  ...
            }
       }
   }

...
Tarea tarea = new Tarea();
tarea.setPriority(1);
tarea.start();
...
```

Utilizando la interfaz Runnable

```
public class Tarea implements Runnable {
       public void run() {
            while (true) {
                  ...
                  this.sleep(1000);
                  ...
            }
      }
}

...
Thread thread = new Thread(new Tarea(),"tarea");
thread.setPriority(10);
thread.start();
...
```