

● Actividad 1:

```
import subprocess

proceso1 = subprocess.Popen(["notepad.exe"]);
proceso2 = subprocess.Popen([r"C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe"]);
proceso3 = subprocess.Popen([r"C:\Program Files
(x86)\Microsoft\Edge\Application\msedge.exe"]);

print("Procesos lanzados:")

print("PID 1:", proceso1.pid);
print("PID 2:", proceso2.pid);
print("PID 3:", proceso3.pid);

proceso1.wait();
proceso2.wait();
proceso3.wait();
print("Los 3 procesos han finalizado");
```

● Actividad 2:

```
public class actividad2 {

    static int contador;

    static class miHilo extends Thread{

        public void run(){
            for(int i=0;i<100000000;i++){
                // contador++ con tan solo esto generaría condición
                de carrera y para solucionarlo utilizamos synchronized
                synchronized (actividad2.class) {
                    contador++;
                }
            }
        }
    }

    public static void main(String[] args) throws
InterruptedException {
    miHilo hilo1 = new miHilo();
    miHilo hilo2 = new miHilo();
    miHilo hilo3 = new miHilo();

    hilo1.start();
    hilo2.start();
    hilo3.start();
}
```

```

        hilo1.join();
        hilo2.join();
        hilo3.join();

        System.out.println("El valor del contador es: " + contador);

    }

}

```

● Actividad 3:

```

import java.io.IOException;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;

public class actividad3 {

    static class hiloCliente extends Thread{

        private Socket cliente;
        public hiloCliente(Socket cliente){
            this.cliente = cliente;
        }
        public void run(){
            try {
                PrintWriter out = new
PrintWriter(cliente.getOutputStream(), true);
                    out.println("Hola soy tu mini servidor");
                    cliente.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public static void main(String[] args) throws IOException {
        ServerSocket servidor = new ServerSocket(8080);
        System.out.println("Servidor Iniciado");
        while(true){
            Socket cliente = servidor.accept();
            System.out.println("Cliente conectado");
            hiloCliente hilo = new hiloCliente(cliente);
            hilo.start();
        }
    }
}

```

}