Segundo Certamen: Tiempo: 15:40 hrs - 17:15 hrs.
Todas las preguntas tienen igual puntaje.

1.- bc es un utilitario de Linux que permite evaluar expresiones matemáticas. Con bc se pueden hacer cosas sofisticadas, pero esta pregunta usted lo usará para evaluar expresiones enteras sencillas como en:

```
agustin@agustin-laptop:~$ echo "3*(5-2)" | bc
9
agustin@agustin-laptop:~$ echo "3*(5-2)+28/((5+2)*2)" | bc
11
```

Con la información previa, se pide que usted haga un programa en Java que pida por consola o una ventana de diálogo (lo que usted desee) ingresar una expresión aritmética al estilo de las previas y entregar su evaluación por consola, luego el programa termina.

```
import java.io.*;
import java.util.Scanner;

class Aritmetica
{
  public static void main(String[ ] args) {
    Runtime runTime= Runtime.getRuntime();
    Process process=null;
    try {
      process = runTime.exec("bc");
      PrintWriter out = new PrintWriter(process.getOutputStream(),
                                        true /* autoFlush */);
      BufferedReader in = new BufferedReader(
        new InputStreamReader(process.getInputStream()));

      BufferedReader console = new BufferedReader(
        new InputStreamReader(System.in));
      String expression = console.readLine();
      out.println(expression);
      System.out.println(in.readLine());
    } catch (IOException e) {
      System.out.println("No pudimos correr el bc");
      System.exit(-1);
    }
    if (process!=null)
      process.destroy();
    try {
      process.waitFor();
    } catch( InterruptedException e ) {
      System.out.println("No pudimos esperar por término");
      System.exit(-1);
    }
    System.out.println("Estatus de término:  "+process.exitValue());
    System.exit(0);
  }
}
```

2.- Se desea medir el tráfico acumulado de subida más bajada entre un cliente único y un servicio de texto cualquiera que corre en el puerto 1234. Suponga que las transferencias son siempre líneas de texto. Para esto se le pide hacer correr el servicio en el puerto 4321 y poner un programa Java en el puerto 1234. Así cuando un cliente se conecta al puerto 1234, su programa hará un puente al 4321. Estando en el medio y **usando un contador único** su programa podrá contar los bytes totales de subida más bajada. Se desea que esta cuenta sea mostrada por pantalla cada vez que llegue un

nuevo paquete en cualquiera de las dos direcciones (cliente → servicio o servicio → cliente). Muestre su programa Java.

Cosas importantes en esta pregunta son: Creación de hebras (una o dos), manejo de sockets, manejo de acceso exclusivo a un contador.

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

public class TrafficMeter
{
   public static void main(String[] args ) {
      try {
         int i = 1;
         ServerSocket s = new ServerSocket(1234);
         Socket incoming = s.accept( );
         Socket outgoing = new Socket ("localhost", 4321);
         Counter counter = new Counter();
         new ThreadedTunnel(incoming, outgoing, counter).start();
         new ThreadedTunnel(outgoing, incoming, counter).start();
      }
      catch (Exception e) {
         e.printStackTrace();
      }
   }
}

class Counter
{
  private int counter;
  public Counter () {
   counter =0;
  }
  public synchronized void add(int i){
    counter+=i;
  }
  public synchronized int getValue() {
    return counter;
  }
}

class ThreadedTunnel extends Thread
{
   public ThreadedTunnel(Socket i, Socket o, Counter c)  {
      incoming = i; outgoing=o; counter = c;
   }

   public void run()  {
      try  {
         BufferedReader in = new BufferedReader
            (new InputStreamReader(incoming.getInputStream()));
         PrintWriter out = new PrintWriter
            (outgoing.getOutputStream(), true /* autoFlush */);
         String str;
         while (true)  {
            str = in.readLine();
           out.println(str);
            counter.add(str.length());
            System.out.println(counter.getValue());
         }
      }
      catch (Exception e)  {
         e.printStackTrace();
      }
   }
```

```
   private Socket incoming, outgoing;
   private Counter counter;
}
```

3.- Se tiene una Base de Datos "Evaluacion" ya registrada con mySql. Suponga el uso de un servidor XAMPP para proveer mySql.
a) A usted se le pide hacer el programa Java CreaTablaProyecto.java. Este programa crea la tabla Proyecto con los campos "rutAlumno" int(11), "E1" int(3), "E2" int(3), hasta "Ei" int(3) con i ingresado como parámetro en la línea de comandos. Ei corresponde a la nota del evaluador i. La tabla debe quedar vacía luego de su creación.
b) Indique ¿qué se requiere además del JDK para que su programa pueda ser compilado? y ejecutado, y ¿Cuáles serían los comandos para compilar y ejecutar su programa?

a)
```
/**
 * Este ejemplo supone que exite la Base de Datos "Evaluacion" ya registrada
 * con MySQL, a la cual se le añade la tabla Proyectos con los campos rutAlumno int(11),
 * E1 int(3) ... hasta Ei int(3) con i ingresado como parámetro en la línea de comandos.
 */

import java.sql.*;

class CreaTablaProyecto {
    static public void main( String[] args ) {
        Connection conexion;
        Statement sentencia;

        try {  // se carga el Driver JDBC
            Class.forName("com.mysql.jdbc.Driver" );
        } catch( Exception e ) {
            System.out.println( "No se pudo cargar el driver" );
            e.printStackTrace();
            return;
        }
        try {
              // Se establece la conexión con la base de datos
              // Se solicita a DriverManager que proporcione una conexión para una
              // fuente de datos MySQL, con nombre "Evaluacion"
            conexion = DriverManager.getConnection("jdbc:mysql://localhost/Evaluacion","root", "");
            System.out.println( "Conexion establecida" );
            sentencia = conexion.createStatement();
            try {
                  // Se elimina la tabla en caso de que ya existiese
                sentencia.execute( "DROP TABLE Proyecto" );
            } catch( SQLException e ) {System.out.println( "Tabla no eliminada" );};
            String SQLstatement="CREATE TABLE Proyecto ( rutAlumno int(11) NOT NULL";
            for (int i=Integer.parseInt(args[0]); i>0; i--)  // the order does not matter
                 SQLstatement += ", E" + i + " int(3) NOT NULL";
            SQLstatement += ")";
            sentencia.executeUpdate( SQLstatement );
            sentencia.close();
            conexion.close();
        } catch( Exception e ) {System.out.println( e ); return;  }
        System.out.println( "Creacion finalizada." );
    }
}
```

b) Para la compilación no se requiere nada más que la JDK. Para ser ejecutado se requiere contar con el driver compatible con el administrador de driver de Java. La ruta completa para ese driver debe ser incluida, en forma adicional a otras posibles rutas, en la variable CLASSPATH o bien incluida en la línea de ejecución con opción -cp
Para compilar: javac CreaTablaProyecto.java
Para ejecutar: java -cp <directorio_del_driver>/<nombre_del_driver>.jar:. CreaTablaProyecto