

Informe Taller 1 (scripts)

Taller de Sistemas Operativos
Escuela de Ingeniería Informática

Fabián Rozas Alfaro

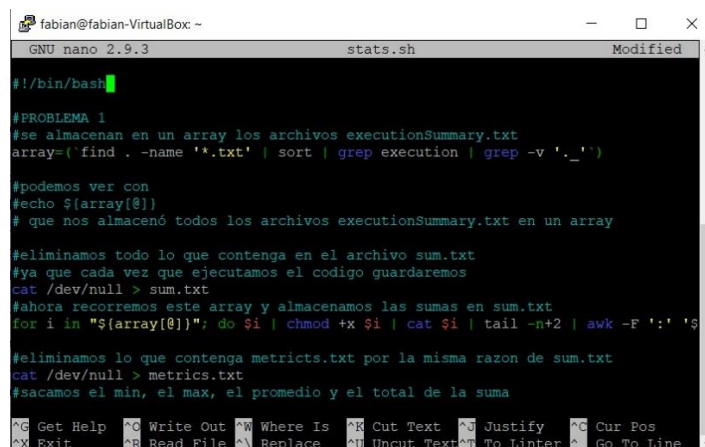
fabian.rozas@alumnos.uv.cl

Resumen. Como objetivo principal del trabajo es realizar una implementación de scripts para procesar resultados de programas basados en un estudio de simulaciones de desempeño de evacuaciones. Este informe muestra la descripción del problema y el diseño de una solución en el cual nos basaremos para su implementación con scripts.

1. Introducción

Para llevar a cabo la implementación utilicé un software de virtualización llamado VirtualBox, el cual tenía instalado con anterioridad en mi computador. Creé una máquina virtual con el sistema operativo Ubuntu, el cual se basa en el uso de comandos de Linux.

Un scrip es un archivo que contiene un conjunto de órdenes para realizar una acción. Para generar un scrip se utilizan lenguajes interpretados, en los cuales se pueden escribir el código que compondrá el script. Este código no será más que un fichero de órdenes o procesamiento por lotes [1]. En este caso ocuparemos como lenguaje interpretado el propio **BASH** (Bourne-again shell), que básicamente es un lenguaje de comandos y shell de Unix. En otras palabras es un procesador de comandos que generalmente se ejecuta en una ventana de texto como lo muestra la Figura 1, donde el usuario escribe comandos que causan acciones. Dicho anteriormente este también puede leer y ejecutar comandos desde un archivo, llamado script de shell [2].



```
fabian@fabian-VirtualBox: ~  
GNU nano 2.9.3 stats.sh Modified  
#!/bin/bash  
  
#PROBLEMA 1  
#se almacenan en un array los archivos executionSummary.txt  
array=(`find . -name '*.txt' | sort | grep execution | grep -v '._'`)  
  
#podemos ver con  
#echo ${array[@]}  
# que nos almacenó todos los archivos executionSummary.txt en un array  
  
#eliminamos todo lo que contenga en el archivo sum.txt  
#ya que cada vez que ejecutamos el código guardaremos  
cat /dev/null > sum.txt  
#ahora recorremos este array y almacenamos las sumas en sum.txt  
for i in "${array[@]"; do $i | chmod +x $i | cat $i | tail -n2 | awk -F ':' '{ $  
  
#eliminamos lo que contenga metrics.txt por la misma razón de sum.txt  
cat /dev/null > metrics.txt  
#sacamos el min, el max, el promedio y el total de la suma
```

Figura 1. Ventana de texto con comandos

Los script tienen encabezamientos conocidos como shebang según el intérprete usado, que le indica al sistema cual ocuparemos. En este caso estamos trabajando con Linux, y ocuparemos el intérprete de Bash. El shebang que se utiliza para este intérprete es `#!/bin/bash` (Figura2), pero en cualquier otro caso será el binario que apunta hacia el intérprete o shell. Además, el fichero de script suele tener la extensión “.sh”. [3]

```
#!/bin/bash
#PROBLEMA 1
#se almacenan en un array los archivos executionSummary.txt
array=(`find . -name '*.txt' | sort | grep execution | grep -v '._'`)
```

Figura 2

2. Descripción del problema

Dicho anteriormente el problema consiste en realizar una implementación de scripts en Bash, el cual debe de permitir procesar los resultados estadísticos basados en un estudio de simulación de un sistema de evacuaciones de personas. El trabajo se compone de tres problemas, que consisten en realizar algunos calculos del promedio, mínimo y máximo, en cada uno de los archivos respectivos a las simulaciones.

2.1 Contexto de los datos a utilizar

Los datos corresponden a estadísticas que provienen de cierta cantidad de experimentos de simulación de evacuación de personas. Se consideran a 75000 personas, las conforman tres modelos: Residentes, Visitantes Tipo I y Visitantes Tipo II (Tabla 1).

<i>Tipo de Persona</i>	<i>Descripción</i>
Residente	Persona que vive en la ciudad y conoce su zona segura
Visitante Tipo I	Visitante que durante la evacuación logra determinar su zona segura
Visitante Tipo II	Visitante que durante la evacuación logra no determinar su zona segura

Tabla 1. Tipos de personas simuladas

Además de la clasificación mencionada, cada persona pertenece a un grupo etario, los que se resumen en la Tabla 2.

<i>Grupo Etario</i>	<i>Intervalo de edad</i>
G0	0-14
G1	15-29
G2	30-64
G3	65 o más

Tabla 2. Grupos etarios

2.2 Organización de los datos

Cada simulación entrega los resultados ordenados en una estructura de directorios, según el esquema que se visualiza en la Figura 3, donde NNN es el identificador de la simulación, representado por un código numérico de tres dígitos, como por ejemplo el 000. En el directorio stats, se almacenan tres archivos de texto (*executionSummary-NNN.txt*, *summary-NNN.txt*, *usePhone-NNN.txt*), los cuales almacenan los datos importantes de cada simulación.

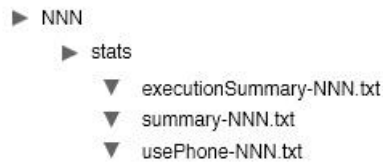


Figura 3. Estructura de directorios de los resultados de cada simulación

2.3 Descripción de los datos

2.3.1 Archivo executionSummary.txt

El archivo executionSummary-NNN.txt contiene datos sobre el desempeño de la simulación NNN. Posee una cabecera con campos del nombre que representa cada dato, y una segunda línea con los datos. Se compone por 10 campos, separados por el símbolo ':' como se muestra en la Figura 4. También en la Tabla 3 se detallan los campos, su descripción y un ejemplo de los datos.

```
numExperiment:tsim:calibrationTime:Residents:Visitors:timeExecMakeAgents:timeExecCal:timeExecSim:maxMemory:agentsMem  
0:3600:100:69000:6000:28252:33157:182800:288236:33648
```

Figura 4. Estructura del archivo executionSummary.txt

<i>Campo</i>	<i>Descripción</i>	<i>Ejemplo</i>
numExperiment	Número del experimento.	0 (Corresponde al código 000)
Tsim	Tiempo de simulacion	3600: segundos de tiempo real que se simuló.
CalibrationTime	Tiempo de calibración del simulador	100: segundo de tiempo de real que se destina a la calibración inicial de las personas.
Residents	Cantidad de Residentes simulados	69000
Visitors	Cantidad de Visitantes simulados	6000

timeExecMakeAgents	Tiempo real que demora en crear en memoria las personas simuladas	28252: milisegundos
timeExecCal	Tiempo real que demora la calibracion de las personas	33157: milisegundos
timeExecSim	Tiempo real que toma la simulación	182800: milisegundos
maxMemory	Costo espacial del simulador	288236: Kbytes
agentsMem	Memoria utilizada por las estructuras de datos relacionados con las personas	33648: KBytes

Tabla 3. Descripción de los campos del archivo executionSummary

2.3.2 Archivo summary.txt

El archivo summary-NNN.txt contiene datos sobre el comportamiento de las personas en la simulación NNN. Posee una cabecera, seguida de tantas líneas como la cantidad de personas que se simularon. Ésta líneas tienen ocho campos, separados por el símbolo ':'. En la Figura 5 se muestra el contenido del archivo que corresponde a la simulación 000. En la Tabla 4 se detallan los campos, su respectiva descripción y un ejemplo.

```

humExperiment: id: model: groupAge: safeZone: distanceToTargetPos: responseTime: evacTime
0:0:0:2:Z1:8.271001:262.785961:1450.000000
0:1:0:1:Z2:7.345218:287.944671:1280.000000
0:2:0:0:Z5:17.635108:82.167863:1230.000000
0:3:0:1:Z1:23.365319:258.646227:2340.000000
0:4:0:2:Z3:5.798595:176.579702:1360.000000
0:5:0:2:Z5:28.269759:139.817949:1300.000000
0:6:0:1:Z1:13.396806:246.938400:2340.000000
0:7:0:2:Z2:4.343360:275.877914:1650.000000
0:8:0:2:Z1:2.693826:20.597654:1480.000000
0:9:0:2:Z2:3.609671:193.424042:2260.000000
0:10:0:0:Z5:6.328705:239.016398:1070.000000
0:11:0:1:Z3:11.706089:138.799325:1840.000000
0:12:0:0:Z2:5.255669:53.876555:1610.000000
0:13:0:0:Z3:26.870673:128.760384:1680.000000
0:14:0:2:Z5:3.673287:53.881971:1180.000000
0:15:0:2:Z1:13.871407:209.411742:2300.000000
0:16:0:2:Z5:0.444277:205.829894:1600.000000
0:17:0:0:Z2:0.720502:443.053518:1420.000000
0:18:0:2:Z5:43.338012:281.720581:1550.000000
0:19:0:0:Z1:12.921273:176.665495:2210.000000
0:20:0:1:Z2:17.608086:199.888670:1760.000000
0:21:0:3:Z2:803.456131:206.926582:0.000000
0:22:0:2:Z2:0.981519:113.177722:2330.000000
0:23:0:0:Z4:51.234965:310.788155:1790.000000
0:24:0:3:Z5:0.269871:186.751243:660.000000
0:25:0:2:Z1:6.004125:466.744458:1530.000000
0:26:0:0:Z2:18.537571:106.474751:1880.000000
0:27:0:0:Z5:17.581589:435.879711:1090.000000
0:28:0:1:Z2:32.201460:170.972578:1400.000000
0:29:0:2:Z3:10.190274:88.030609:1290.000000
0:30:0:2:Z1:16.642132:239.112689:1710.000000
0:31:0:2:Z2:18.308093:104.497766:1220.000000
0:32:0:1:Z5:29.540280:179.253064:1210.000000
0:33:0:0:Z2:4.142672:292.593174:2150.000000
0:34:0:2:Z4:8.426183:260.742798:1320.000000
0:35:0:0:Z3:16.503360:164.005737:1510.000000
0:36:0:0:Z1:62.378806:157.433456:1970.000000
0:37:0:3:Z4:67.041733:78.937755:2560.000000
0:38:0:2:Z4:24.287050:163.673256:950.000000
0:39:0:2:Z5:14.288496:111.308010:1230.000000
0:40:0:3:Z1:197.678312:323.091362:3350.000000

```

Figura 5. Estructura archivo summary.txt.

<i>Campo</i>	<i>Descripción</i>	<i>Ejemplo</i>
numExperiment	Número del experimento.	0 (Corresponde al código 000)
id	Identificador de la persona simulada	40
model	Identificar del modelo de la persona 0: Residente 1: Visitante tipo I 2: Visitante tipo II	0
gruoupAge	Identificador del grupo etario de la persona 0: G1, 1: G2, 2: G3, 3: G4	3
safeZone	Identificador de la zona segura de la persona Z1, Z2, Z3, Z4, Z5 : Identificadores de zonas seguras NA: la persona no tiene zona segura asignada	Z1
distanceToTargetPos	Distancia a la que quedó la persona de su objetivo inicial	197.678312 metros
responseTime	Número aleatorio que representa cuánto tiempo se demoró la persona en tomar la decisión de evacuar, desde que dieron la orden de evacuar. Si es -1, entonces la persona nunca pudo evacuar.	323.091362 segundos
evacTime	Tiempo que la persona se demoró en llegar a la zona de evacuación. Si es 0, entonces la persona nunca llegó a la zona de evacuación.	3350 segundos

Tabla 4 Descripción de los campos del archivo summary.txt

2.3.3 Archivo usePhone.txt

El archivo usePhone-NNN.txt contiene datos sobre el comportamiento de las personas en la simulación NNN, con respecto al uso de un teléfono móvil. Posee una cabecera y cada línea posterior, representa la cantidad de personas que utilizaron el teléfono en cierto instante de tiempo. Cada una tiene tres campos, separados por el símbolo ‘:’. En la Figura 6 se muestra el contenido del archivo que corresponde a la simulación 000. En la Tabla 4 se detallan los campos, su respectiva descripción y un ejemplo.

```
|usePhone
0:0:0
0:10:4333
0:20:4383
0:30:4261
0:40:4410
0:50:4388
0:60:4381
0:70:4440
0:80:4367
0:90:4506
0:100:4436
0:110:4364
0:120:4542
0:130:4368
0:140:4394
0:150:4426
0:160:4405
0:170:4504
0:180:4459
0:190:4410
0:200:4423
0:210:4422
0:220:4273
0:230:4242
0:240:4266
0:250:4114
0:260:4018
0:270:4143
0:280:3949
0:290:3951
0:300:3791
0:310:3686
0:320:3611
0:330:3431
0:340:3427
0:350:3365
0:360:3292
0:370:3142
0:380:3020
0:390:2825
0:400:2844
```

Figura 6 Estructura archivo usePhone.txt

<i>Campo</i>	<i>Descripción</i>	<i>Ejemplo</i>
numExperiment	Número del experimento	0 (Corresponde al código 000)
timeStamp	Tiempo de la medición	30
usePhone	Cantidad de persona que utilizaron el teléfono móvil en el tiempo especificado	4261

Tabla 5 Descripción de los campos del archivo usePhone.txt.

3. Diseño de la solución

El diseño consiste básicamente en el análisis y posterior funcionamiento del script, basado en los requerimientos de cada uno de los problemas planteados en el taller 1. Está debe de poder lograr un orden de los comandos de Linux, realizando cada una de los problemas requeridos y finalmente una implementación exitosa.

Los problemas son los siguientes:

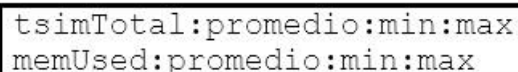
- Problema 1: Se necesita el tiempo de simulación y la memoria utilizada por la simulación.
- Problema 2: Se necesitan los cálculos de todas las personas simuladas, Residentes, Visitantes tipo I, Residentes separados por grupo etario y Visitantes tipo I separados por grupo etario.
- Problema 3: Se necesita el cálculo del uso de teléfonos en el instante de tiempo específico.

Se utilizó un diseño similar para los 3 problemas, puesto que se pide buscar el total de estos datos y poder calcular el promedio, max y el min en cada uno de ellos.

Primero que todo, se deben de buscar los archivos descargados de cada simulación y almacenarlos en tres array distintos (uno para todos los archivos *executionSummary-NNN.txt*, otro para *summary-NNN.txt*, y otro para *usePhone-NNN.txt*). Después se recorren los array, buscando y realizando la sumas, o extracción de datos respectivos.

En el primer problema, se realiza la extracción de las sumas que se componen por los campos de *timeExecMakeAgents*, *timeExecCal* y *timeExecSim*, que se encuentran en el array dónde están almacenados todos los archivos de *executionSummary-NNN.txt*. Los resultados de las sumas se almacenan en un nuevo archivo. Estas sumas representan el tiempo de simulación total para cada *executionSummary-NNN.txt*. Luego para sacar el min, el max y el promedio visualizamos el tiempo de simulación total de cada una de las simulaciones almacenadas en el nuevo archivo. Finalmente al obtener el min, max, promedio y la suma total del tiempo de todas las simulaciones, se almacenan en un archivo llamado '*metrics.txt*'.

Ahora para sacar el min, max, promedio y suma total de la memoria utilizada por las simulaciones, se utiliza el mismo diseño anterior. Se busca en el array la columna o campo que representa la memoria utilizada, y realizando las sumas correspondientes para almacenarlas en un nuevo archivo. Luego, se calcula el promedio, max y min en dicho archivo. Para almacenar los resultados en el archivo '*metrics.txt*'. Finalmente, quedando en el archivo *metrics.txt* algo como se muestra en la Figura 7.



```
tsimTotal:promedio:min:max
memUsed:promedio:min:max
```

Figura 7. Problema 1

En el segundo problema se debe calcular las sumas del tiempo de evacuación de todas las personas simuladas. Dicho campo se encuentra en los archivos de *summary-NNN.txt*, que como dicho anteriormente se almacenaron en un array. Se extraen los campos necesarios para realizar las sumas del tiempo de

evacuación y se almacenan en un nuevo archivo. Luego, para calcular el total, promedio, min y max se debe entrar en el nuevo archivo, realizar las operaciones correspondientes y almacenar los resultados en 'evacuation.txt'. Así mismo se deben de encontrar el total, promedio, min y max del tiempo de evacuación, pero de los siguientes grupos o tipos de personas simuladas, y también almacenar los resultados en el archivo *evacuation.txt*. Finalmente quedando compuesto dicho archivo como se muestra en la Figura 8.

```
alls : promedio: min:max  
residents :promedio: min:max  
visitorsI :promedio: min:max  
residents-G0 :promedio: min:max  
residents-G1 :promedio: min:max  
residents-G2 :promedio: min:max  
residents-G3 :promedio: min:max  
visitorsI-G0 :promedio: min:max  
visitorsI-G1 :promedio: min:max  
visitorsI-G2 :promedio: min:max  
visitorsI-G3 :promedio: min:max
```

Figura 8

Para el último problema, al igual que los dos anteriores, se pide encontrar y calcular el promedio, min y max. Pero, en este caso se basan en el campo del uso de teléfonos móviles para cada uno de los instantes de tiempo especificados en los archivos de *usePhone-NNN.txt* almacenados en el array correspondiente. Es decir, que se deben encontrar la cantidad de personas que utilizaron el teléfono en un determinado tiempo de cada simulación, y almacenarlos en un archivo. Luego para calcular el promedio, el max y el min en un tiempo determinado, se debe de buscar en dicho archivo todos los resultados en el determinado tiempo, calculando el promedio de estos, el min y el max. Para finalmente almacenar los resultados en un archivo llamado 'usePhone-stats.txt'. En la Figura 9 se muestra un ejemplo del archivo *usePhone-stats.txt* en un instante de tiempo.

```
timestamp:promedio:min:max
```

Figura 9

3.2 Diagrama de alto nivel

En la Figura 10 se muestra el diagrama relacionado al diseño de la solución.

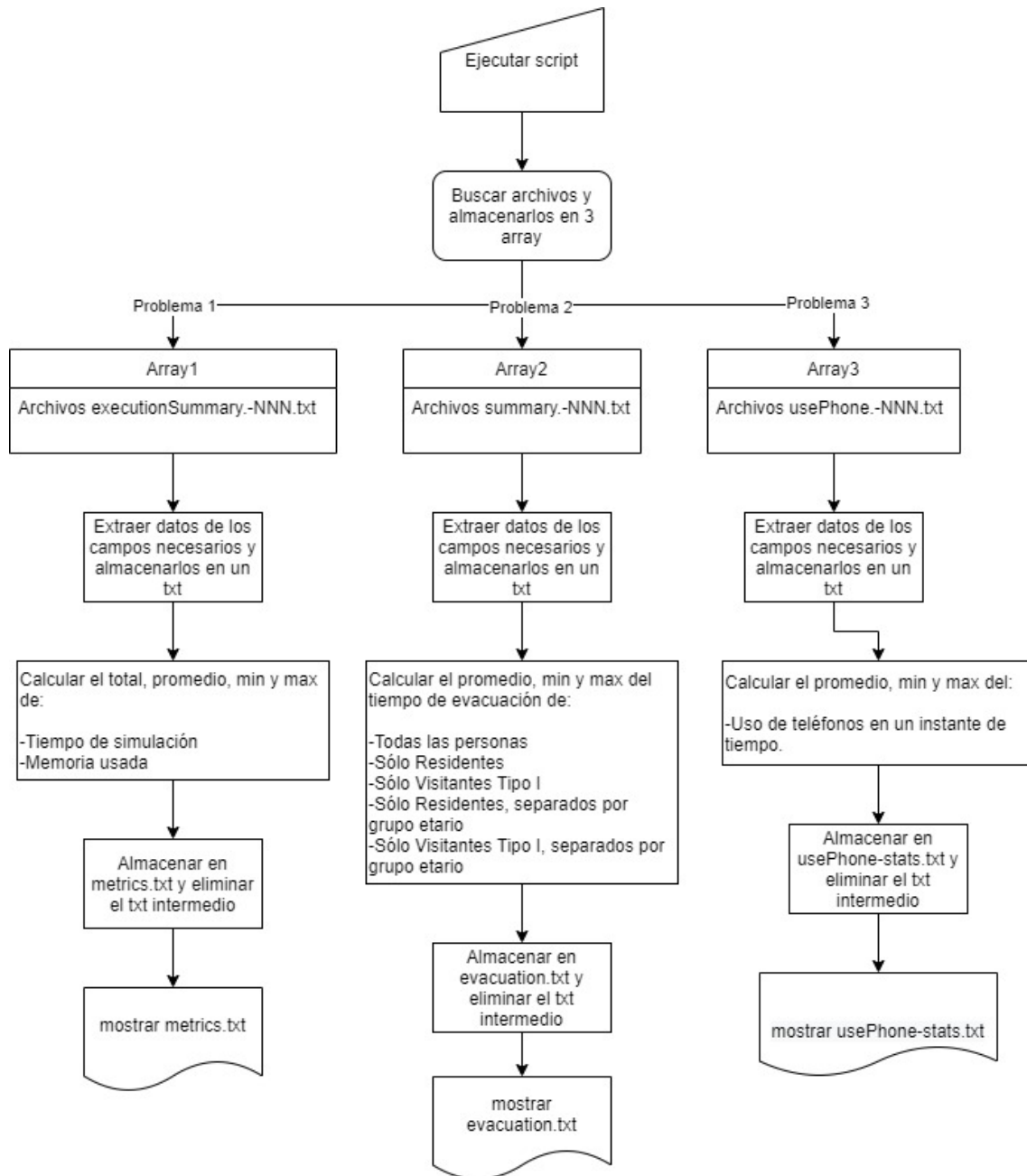
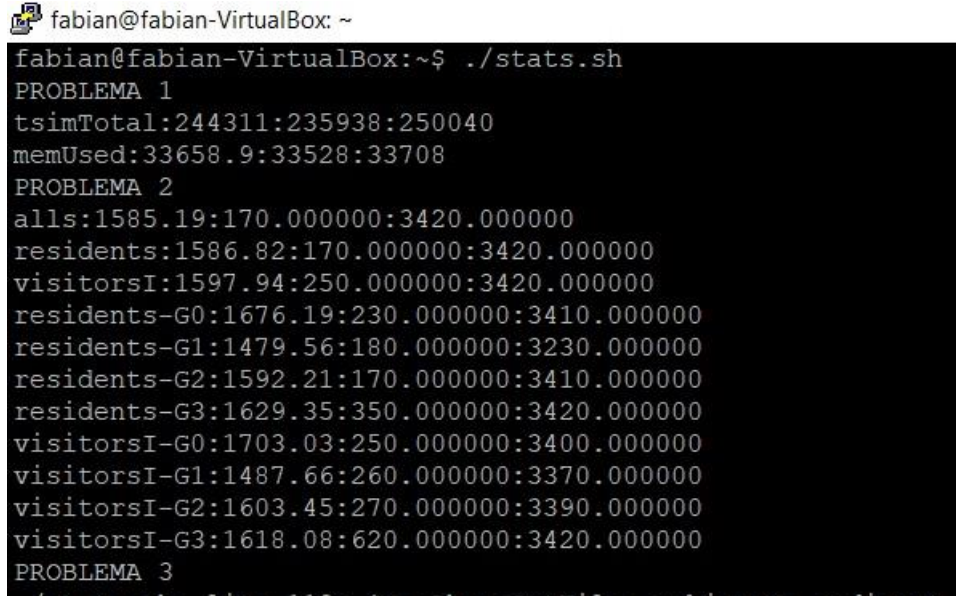


Figura 10. Diagrama de la solución

Se puede observar que cada problema tiene una solución similar. También cabe mencionar que al mostrar el primer problema, comienza con la ejecución del siguiente.

4. Resultados

Se implementó el script, y se obtuvieron los resultados de cada uno de los problemas. En la Figura 11 se puede apreciar con mas detalle lo que nos arrojó la consola con el comando para ejecutar el script “./stats.sh”.



```
fabian@fabian-VirtualBox: ~  
fabian@fabian-VirtualBox:~$ ./stats.sh  
PROBLEMA 1  
tsimTotal:244311:235938:250040  
memUsed:33658.9:33528:33708  
PROBLEMA 2  
alls:1585.19:170.000000:3420.000000  
residents:1586.82:170.000000:3420.000000  
visitorsI:1597.94:250.000000:3420.000000  
residents-G0:1676.19:230.000000:3410.000000  
residents-G1:1479.56:180.000000:3230.000000  
residents-G2:1592.21:170.000000:3410.000000  
residents-G3:1629.35:350.000000:3420.000000  
visitorsI-G0:1703.03:250.000000:3400.000000  
visitorsI-G1:1487.66:260.000000:3370.000000  
visitorsI-G2:1603.45:270.000000:3390.000000  
visitorsI-G3:1618.08:620.000000:3420.000000  
PROBLEMA 3  
./stats.sh: line 116: Syntax error: Unterminated quoted string
```

Figura 11

5. Conclusión

Ya mencionado anteriormente, se puede observar que se utilizaron procesos similares para desarrollar cada problema. El diagrama de alto nivel resultó de mucha ayuda debido al orden de sus procesos, los cuales permitieron conformar una implementación mucho más simple y bien desarrollada del script. Como último punto, el objetivo del taller fue logrado exitosamente, ya que se aprendieron algunos de los comandos más importantes de BASH en Linux, lo que nos permite a futuro poder manejarnos y trabajar de forma mas rápida en este sistema operativo.

6. Referencias

- [1] ¿Qué es un script?, <https://www.linuxadictos.com/que-es-script.html>
- [2] Bash, 21 de Mayo 2020, <https://es.wikipedia.org/wiki/Bash>
- [3] <https://www.linuxadictos.com/que-es-script.html>