

# Informe Taller 1

Taller de Sistemas Operativos  
Escuela de Ingeniería Informática

Abel Núñez Cataldo

Abel.nunez@alumnos.uv.cl

**Resumen.** Este informe documenta los procedimientos que se llevaron a cabo sobre el estudio de simulaciones de desempeño de evacuaciones en la ciudad de Iquique, y el diseño de una solución que permite obtener estadísticas descriptivas, basadas en los resultados de estas simulaciones, utilizando las herramientas proveídas en clase.

## 1. Introducción

Bash es una herramienta de desarrollo de software que trabaja directamente sobre el sistema operativo, agilizando la utilización de memoria y procesamiento de código. Para llevar a cabo este ejercicio se utilizó, el software de virtualización distribuido por Oracle llamado VirtualBox, con el cuál se instaló una máquina virtual de Ubuntu Server versión 18.04, que se basa principalmente en el uso de comandos Linux.

El objetivo de este taller, consiste en realizar el diseño e implementación de una solución, que permita el estudio y manejo de datos, referidos de distintas simulaciones y generar estadísticas descriptivas, en este caso las simulaciones se contextualizan en la evacuación costera de la ciudad de Iquique ante un eventual Tsunami. Para llevar esto a cabo es necesario implementar scripts, en lenguaje de comandos Bash, para procesar los datos entregados por las simulaciones y almacenar los resultados en documentos de salida, por cada requerimiento del taller.

Este documento cuenta con una introducción que abarca la tecnología utilizada, los objetivos a alcanzar. Seguida de la descripción del problema planteado en el taller, que incluye la descripción de los datos utilizados, el significado de las variables utilizadas y algunos ejemplos en que se instancian. Finalizando con el diseño de la solución, en la que se muestran diagramas de alto nivel y los métodos utilizados para la solución del problema.

## 2. Descripción del problema

El problema establecido en este taller consiste en diseñar un script en Bash, que permita realizar estadísticas descriptivas de los datos entregados por las simulaciones. El trabajo se divide en 3 situaciones, que se considerarán como “Tareas”, que consisten en realizar prácticamente los mismos cálculos, promedio, mínimo y máximo, pero en distintos archivos, con distintas condiciones.

- Tarea 1: Tiempo de simulación y memoria utilizada por la simulación (Archivo destino: “metrics.txt”).
- Tarea 2: Todas las personas simuladas, solo Residentes, solo Visitantes tipo I, solo Residentes separados por grupo etario, solo Visitantes tipo I separados por grupo etario (Archivo destino: “evacuation.txt”)-

- Tarea 3: Uso del teléfono en el instante de tiempo específico (Archivo destino: “usePhone-stats.txt”)

## 2.1 Contexto de los datos a utilizar

Los datos a procesar corresponden a estadísticas provenientes de cierta cantidad de experimentos de simulación de evacuación de personas. Existen 3 modelos de personas: Residentes, Visitantes Tipo I y Visitantes Tipo II (Tabla 1).

Tabla 1 Tipos de personas simuladas

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

Además de esta clasificación, cada persona pertenece a un grupo etario (Tabla 2).

Tabla 2 Grupos etarios utilizados

Grupo Estario	Intervalo de edad
<b>G0</b>	0 – 14
<b>G1</b>	15 – 29
<b>G2</b>	30 – 64
<b>G3</b>	65 o más

## 2.2 Organización de los datos

Cada simulación entrega los resultados ordenados en una estructura de directorios, donde cada directorio cuenta con 3 archivos de texto plano; *executionSummary-nnn.txt*, *summary-NNN.txt* y *usePhone-NNN.txt*, los cuales contienen los datos, separados por el símbolo “:” y nos permitirán el análisis para el taller, según el esquema que se muestra en la Figura 1, donde NNN es el identificador de la simulación, representada por un código de tres dígitos.

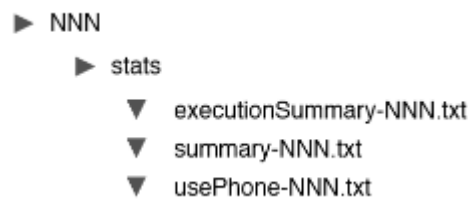


Figura 1 Estructura de directorios de los resultados de cada simulación.

## 2.3 Descripción de los datos

### 2.3.1 Archivo executionSummary.txt

Los archivos *executionSummary-NNN.txt* contienen datos sobre el desempeño de la simulación NNN. Posee una cabecera y una segunda línea con los datos. Ésta consta de 10 campos. En la Tabla 3 se detallan los campos y su descripción.

Tabla 3 Descripción de los campos de executionSummary.txt

Campo	Descripción
<b>numExperiment</b>	Número del experimento.
<b>Tsim</b>	Tiempo de simulación.
<b>CalibrationTime</b>	Tiempo de calibración del simulador.
<b>Residents</b>	Cantidad de Residentes simulados.
<b>Visitors</b>	Cantidad de Visitantes simulados.
<b>timeExecMakeAgents</b>	Tiempo real que demora en crear en memoria las personas simuladas.
<b>timeExecCal</b>	Tiempo real que demora la calibración de las personas.
<b>timeExecSim</b>	Tiempo real que toma la simulación .
<b>maxMemory</b>	Costo espacial de simulación.
<b>agentsMem</b>	Memoria utilizada por las estructuras de datos relacionados con las personas.

### 2.3.2 Archivo summary.txt

Los archivos *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. Éstas constan de 8 campos. En la Tabla 4 se detallan los campos y su descripción.

Tabla 4 Descripción de los campos de summary.txt

Campo	Descripción
<b>numExperiment</b>	Número del experimento.
<b>id</b>	Identificador de la persona.
<b>model</b>	Identificador del modelo. 0: Residente; 1: Visitante tipo I; 2: Visitante tipo II

<b>groupAge</b>	Identificador del grupo etario. 0: G1; 1: G2; 2:G3; 3: G4
<b>safeZone</b>	Identificador de la zona segura. Z1, Z2, Z3, Z4, Z5 : Identificadores de zonas seguras. NA: La persona no tiene zona segura asignada.
<b>distanceToTargetPos</b>	Distancia a la que quedó la persona de su objetivo inicial.
<b>responseTime</b>	Número aleatorio que representa cuánto tiempo se demoró la persona en tomar la decisión de evacuar, desde que dieron la orden de evacuación. Si es -1, entonces la persona nunca pudo evacuar.
<b>evacTime</b>	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.

### 2.3.3 Archivo usePhone.txt

Los archivos *usePhone-NNN.txt* contiene datos sobre el comportamiento de las personas en la simulación NNN, con respecto al uso de teléfonos móviles. Posee una cabecera y cada línea posterior, representa la cantidad de personas que utilizaron el teléfono en cierto instante de tiempo. Cada una tiene 3 campos. En la Tabla 4 se detallan los campos y su descripción.

Tabla 5 Descripción de los campos de usePhone.txt

<b>Campo</b>	<b>Descripción</b>
<b>numExperiment</b>	Número del experimento.
<b>timeStamp</b>	Tiempo de mediación.
<b>usePhone</b>	Cantidad de personas que utilizaron el móvil en el tiempo especificado.

## 3. Diseño de la solución

### 3.1 Metodología

La metodología consiste en analizar los requerimientos, diseñar un modelo del funcionamiento del script, donde se pueda ver un orden de los procesos a realizar para lograr el cálculo de las estadísticas requeridas por cada tarea. Una vez estudiado el funcionamiento general de la solución, se debe crear un modelo de diseño más específico para los módulos del taller, es decir, para cada tarea es necesario realizar un diseño de comportamiento, ya que cada archivo presenta una estructura distinta y distintas formas de calcular los resultados.

### 3.2 Diseño

Para el diseño de la solución primero se procede a analizar el problema, estableciendo los siguientes requerimientos:

- Debe poder ingresar como parametro el directorio de la simulación.
- Reconocer el parametro efectivamente como un directorio.
- Debe acceder a los datos de los archivos.
- Calcular las estadísticas descriptivas (mínimo, máximo y promedio) de los datos requeridos.
- Guardar los resultados obtenidos en los archivos de destino correspondientes, con la estructura requerida.

Seguido por el diseño de un diagrama de alto nivel (en este caso un diagrama de flujo, Figura 2) que ayude a explicar el funcionamiento general de la posible solución para el problema.

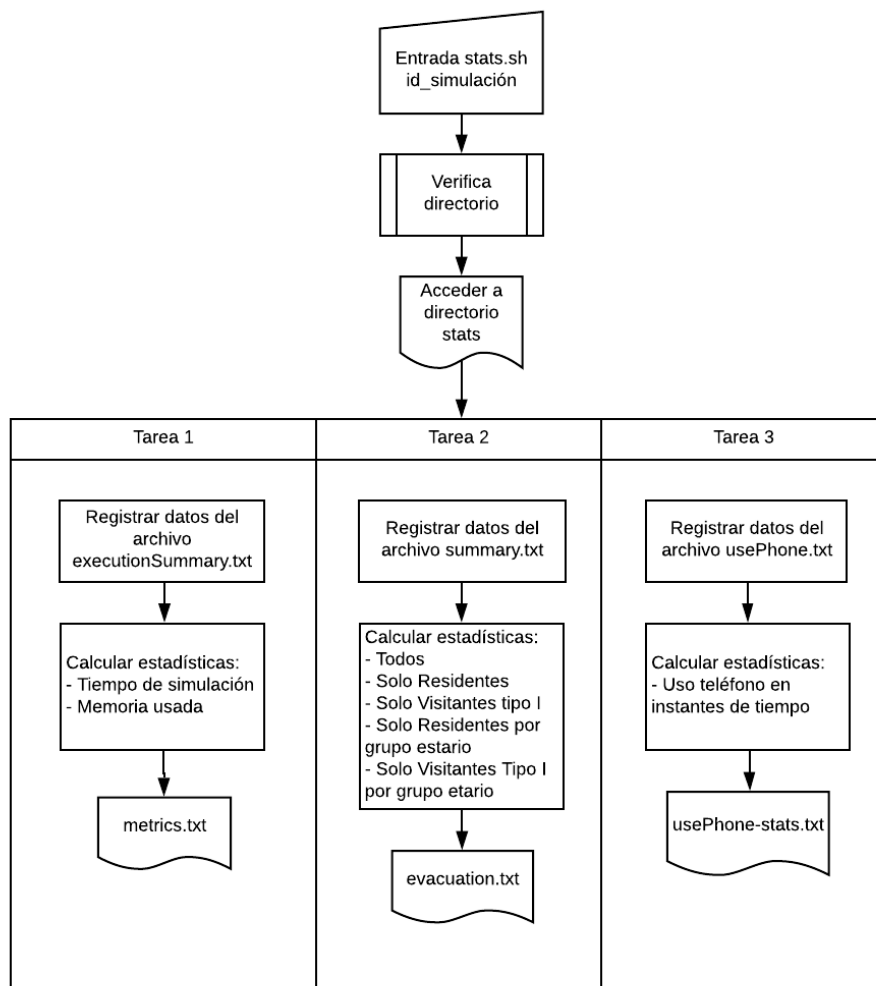


Figura 2 Diagrama de flujo funcionamiento general.

Si bien el diagrama muestra las 3 tareas, lo primero que se debe notar al momento de ejecutar el script, es que este debe recibir como parámetro *id\_simulación*, esto apunta a que debe recibir el identificador de la simulación, y verificar que es efectivamente un directorio, y luego acceder al directorio *stats* que se encuentra dentro. Seguido de esto se registran los datos de los archivos dentro de la simulación que se dió de parámetro, para el cálculo de las estadísticas requeridas. En el caso de la Tarea 1, se debe registrar el contenido del archivo *executionSummary.txt*, para calcular el tiempo de simulación total (*timeExecMakeAgents* + *timeExecCal* + *timeExecSim*), promedio, mínimo y máximo; memoria utilizada por el simulador (*maxMemory* + *agentsMem*), promedio, mínimo y máximo de la memoria total. Una vez realizado este cálculo los resultados deben ser almacenados en el archivo de salida *metrics.txt*, concluyendo la primera tarea.

De manera general, cada una de estas tareas, operan de manera similar, tomar el archivo correspondiente, leer y procesar los datos requeridos, calcular las estadísticas y guardar los resultados en el archivo de salida solicitado. Considerando este procedimiento, es pertinente a la hora de procesar los datos, utilizar archivos temporales que almacenen los datos obtenidos, permitiendo ordenarlos de manera mas clara y comoda, para hacer los cálculos estadísticos, y como estos archivos solo existen durante la ejecución del programa, deben ser eliminados al finalizar la ejecución, dado que no contienen información relevante, en cuanto al resultado final requerido.

#### 4. Resultados

Basado en el diseño anterior se generó el Script, dentro de lo que se pudo lograr, se obtuvieron resultados no satisfactorios con respecto a lo solicitados en el taller debido a que todos los valores obtenidos terminaron siendo los mismos. Esto se puede ver reflejado en la Figura 3, utilizando el comando *more* para mostrar los contenidos de los archivos de salida:

```
abel@abelnc:~/prueba/TSSOO-Taller01$ more metrics.txt
tsimTotal:prom:min:max
248302 : 248302 : 248302 : 248302
memUsed:prom:min:max
324676 : 324676.00 : 324676: 324676
abel@abelnc:~/prueba/TSSOO-Taller01$ more evacuation.txt
alls:promedio:min:max
119057010 119057010 119057010 119057010
residents:promedio:min:max
109541080 109541080 109541080 109541080
visitors1:promedio:min:max
9515930 9515930 9515930 9515930
abel@abelnc:~/prueba/TSSOO-Taller01$ more usePhone-stats.txt
timestamp:prom:min:max
10 : 4475.00 : 0 : 0
abel@abelnc:~/prueba/TSSOO-Taller01$
```

Figura 3 Resultados de la ejecución del script para la simulación 000.

## **5. Conclusiones**

Si bien los resultados no son los correctos, con respecto a los calculos estadísticos de cada elemento de las simulaciones, este sistema bien implementado puede ser de gran utilidad para la extracción de datos, en este tipo de situaciones de evacuación. Con respecto a los objetivos de este taller, se logró aprender sobre nociones básicas de la linea de comandos de Linux “Bash”, y tomar conciencia sobre su complejidad. La cual permite un trabajo directo sobre el sistema operativo y optimizar el rendimiento de la memoria, para ejercicios practicos de desarrollo de scripts.