

Reporte Técnico: Taller 01

Taller de Sistemas Operativos
Escuela de Ingeniería Informática

Fernando Del Pino Machuca

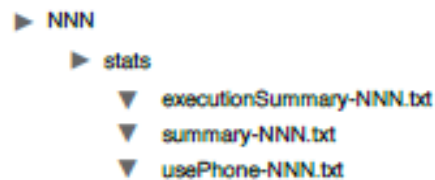
Fernando.delpino@alumnos.uv.cl

Resumen. En este reporte se documenta el manejo de datos almacenados calculando información como promedio, valores mínimos, valores máximos, entre otro tipo de información, con el propósito de implementar scripts para el procesamiento de datos y resultados. Este manejo de información se realizará sobre una serie de archivos de datos, almacenados en subdirectorios, con el propósito de recabar estadísticas sobre experimentos de simulación en un sistema de evacuación de personas. Cada simulación representa una evacuación costera de la ciudad de Iquique ante un eventual tsunami. En base a esto, se maneja información referente a un tipo de residentes y visitantes de la ciudad, información sobre el comportamiento de las personas, tiempos de evacuación, memoria utilizada por el simulador, uso de teléfonos móviles durante instantes de tiempo, entre otros datos relevantes del estudio. Esta información será procesada y se almacenaran las respectivas salidas en los archivos de salida correspondientes.

1. Introducción

Este reporte se basa en el estudio y manejo de datos referentes a un sistema simulado de evacuaciones de personas, en este caso, la simulación se basa en la evacuación costera de la ciudad de Iquique debido a un eventual tsunami en la zona. Los datos de la simulación realizada se encuentran estructurados con un orden específico por cada tipo de resultado estudiado, esta estructura corresponde a 3 archivos de forma general para las 11 simulaciones. En la Fig. 1 se muestra cómo se almacenan los 3 archivos de cada simulación, donde NNN es el identificador de la simulación, representado por un código numérico de tres dígitos.

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



Es necesario destacar la estructura de cada archivo, donde el archivo “executionSummary.txt” está estructurado en la Fig. 2, el archivo “summary.txt” en la Fig. 3 y el archivo “usePhone.txt” en la Fig. 4, junto a sus respectivas tablas explicativas de cada abreviatura.

Figura 2 Estructura de los resultados de la simulación en executionSummary.txt

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

Tabla 1 Descripción de los campos del archivo executionSummary.txt

Campo	Descripción	Ejemplo
numExperiment	Número del experimento	0 (Código de la simulación)
Tsim	Tiempo de simulación	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 calibración 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

Figura 3 Estructura de los resultados de la simulación en summary.txt

```

numExperiment: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

```

Tabla 2 Descripción de los campos del archivo summary.txt

Campo	Descripción	Ejemplo
numExperiment	Número del experimento	0 (Código de la simulación)
id	Identificador de la persona simulada	5

model	Identificar del modelo de la persona 0:Residente 1:Visitante tipo I 2:Visitante tipo II	0
groupAge	Identificador del grupo etario de la persona 0:G1, 1:G2, 2:G3, 3:G4	2
safeZone	Identificadores de 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 se quedó la persona de su objetivo inicial	13.871407 metros
responseTime	Número aleatorio que representa cuanto 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.	209.411742 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	2300 segundos

Figura 4 Estructura de los resultados de la simulación en usePhone.txt

```

numExperiment:timeStamp usePhone
0:0:0
0:10:4333
0:20:4383
0:30:4261
0:40:4410

```

Tabla 3 Descripción de los campos del archivo usePhone.txt

Campo	Descripción	Ejemplo
numExperiment	Número del experimento	0 (Corresponde al código 000)
timeStamp	Tiempo de la medición	60
usePhone	Cantidad de persona que utilizaron el teléfono móvil en el tiempo especificado	4381

En base a esto, el objetivo del taller es la creación de scripts, en lenguaje de comandos “Bash” [1], capaces de procesar la información de estos archivos y entregar resultados en distintos archivos de salida por cada requerimiento del taller. Cada requerimiento del taller se considerará como “Tareas” por hacer. La tarea 1 consiste en determinar la cantidad máxima, mínima, promedio para el tiempo de simulación total y memoria utilizada por el simulador, dentro de los archivos “executionSummary-NNN.txt” de cada simulación, entregando el resultado estructurado y en un archivo de salida “metrics.txt”. La tarea 2 busca determinar el tiempo promedio de evacuación, mínimo y máximo para todas las personas simuladas de primera, luego para Residentes, Visitantes Tipo I, solo Residentes separados por grupo etario, y finalmente solo Visitantes Tipo I

separados por grupo etario, del archivo “summary-NNN.txt” de cada simulación; donde el resultado de estos grupos de personas se almacenará en el archivo de salida “evacuation.txt” con la estructura determinada. Finalmente, la Tarea 3 busca determinar el promedio de uso de teléfonos móviles, valores mínimo y máximo para los instantes de tiempo especificados en cada archivo simulado de “usePhone-NNN.txt”, expresando el resultado estructurado en el archivo de salida “usePhone-stats.txt”. Todas estas tareas deben ser resueltas mediante un solo script con el nombre de “stats.sh”.

Para documentar la creación y implementación del script, este reporte analizara de manera general el funcionamiento correspondiente del script para la toma de archivos de cada directorio, para luego abarcar el procedimiento de la realización de cálculos estadísticos de los datos junto a los resultados entregados por los archivos de salida del programa, donde estos resultados serán concluyentes para el estudio y realización del taller, en cuanto al manejo de comandos mediante script.

Antes de profundizar en el procedimiento de desarrollo del script, es necesario destacar que todo el proceso de desarrollo y manejo de script se realizó a través de la máquina virtual instanciada en estudios previos, la cual se encuentra establecida con sistema operativo Ubuntu (distribución de Linux), la cual permite el uso del lenguaje Bash. El repositorio establecido para esta actividad se clono localmente en esta máquina virtual, junto con el manejo de directorios, archivos y código para interiorizarse más en el manejo del sistema operativo.

2. Procedimiento

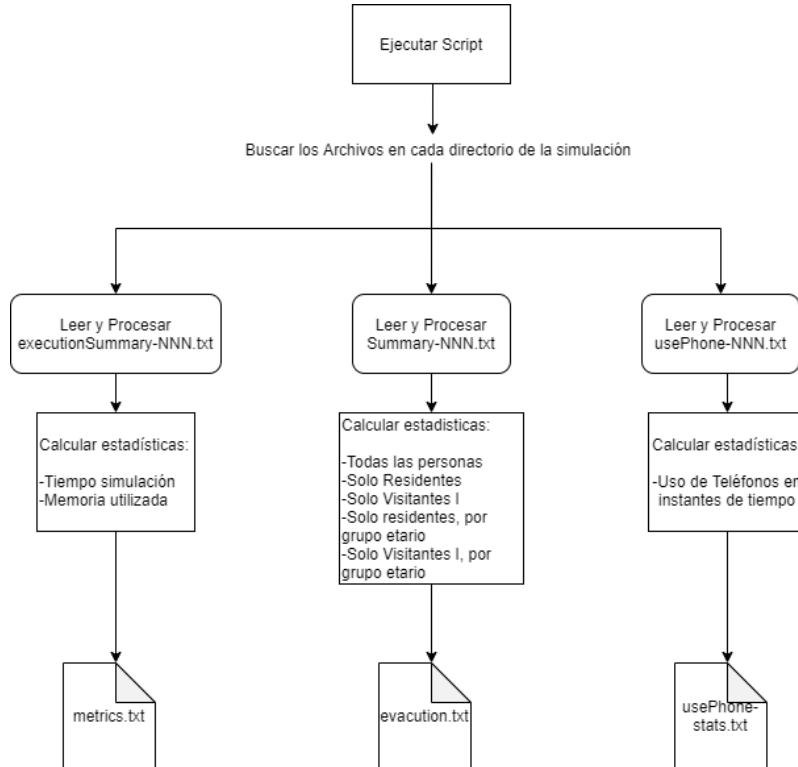
2.1. Metodología:

La metodología para abarcar las tareas propuestas fue analizar cada requerimiento, diseñar un modelo estructural del funcionamiento del script, creando un diseño de vista general de este, donde se vea reflejado un orden de procesos a realizar antes para lograr el cálculo estadístico de los valores requeridos por cada tarea. Una vez analizado el funcionamiento general del script, se debe crear un modelo de diseño específico para cada módulo del taller, eso quiere decir, que se debe idear un diseño de comportamiento para cada tarea, ya que cada archivo presenta una estructura y modo de calcular cada resultado de manera distinta, por lo que una vez especificada cada tarea a realizar por cada módulo en el diseño, se procederá a utilizar como ejemplo para su implementación en código y correspondientemente a iterar frente a posibles errores, hasta encontrar los resultados esperados.

2.2. Diseño:

Para lograr un correcto entendimiento del proceso de toma de datos y cálculos estadísticos que se llevaran a cabo más adelante, es necesario, que se entienda el proceso de forma general, ya que el código debe de poseer varias características sobre procesamiento y lectura de varios archivos estructurados en distintos directorios, además de poseer una larga lista de contenido cada cierto archivo. Este modelo general se ve reflejado en la Fig. 5.

Figura 5. Diseño general de la solución



Como se puede observar, lo primero a la hora de ejecutar el script, es tomar cada archivo almacenado dentro de los directorios de cada simulación, en este caso se recorren 11 archivos de cada estudio de la simulación, donde estos archivos se someten a un proceso de lectura de los datos, procesamiento y toma de elementos específicos para calcular las estadísticas requeridas. En el caso de la Tarea 1, se requiere leer y procesar el archivo “executionSummary.txt” donde este tras tomar los elementos para calcular las estadísticas de tiempo de simulación total($\text{timeExecMakeAgents} + \text{timeExecCal} + \text{timeExecSim}$), promedio, mínimo y máximo del tiempo de simulación total; memoria utilizada por el simulador(agentsMem), promedio, mínimo y máximo de la memoria total. Una vez realizado este cálculo se debe almacenar esta información en el archivo de salida “metrics.txt”, completando así, la primera tarea.

Cada una de estas tareas, de manera general, busca operar de la misma forma, tomar los archivos correspondientes, leerlos, procesar los datos, calcular estadísticas y guardar los resultados dentro de un archivo de salida. En base a este proceso, es conveniente considerar que a la hora de realizar el procesamiento de datos, sea conveniente utilizar archivos temporales para ir almacenando todos los datos obtenidos por cada archivo, ordenando de manera más clara y cómoda los datos para realizar el cálculo estadístico, donde estos archivos temporales que solo existen durante la ejecución del programa, se eliminarán una vez terminada la ejecución del script, por no contener información relevante con respecto a los resultados esperados y requeridos.

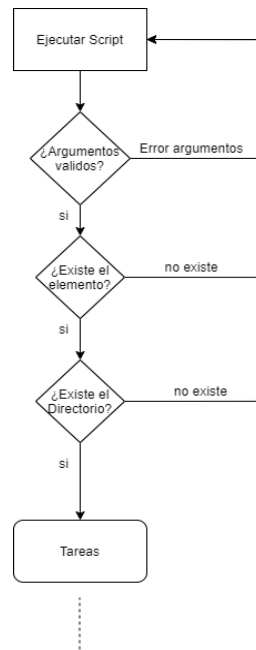
2.3. Implementación:

Una vez realizado la estructuración del diseño de la solución y como se debe comportar el script, procederemos a su creación. Con esto último en mente se procederá a crear un diseño especificado para cada tarea, donde se utilizará como modelo para mayor entendimiento y implementación del funcionamiento del código.

2.3.1. Toma de Datos:

Para comenzar con esta tarea, se buscó en un principio que el programa tomara cada archivo y documento específico del directorio de simulaciones, en base a esto, el programa debería ser capaz de buscar en la carpeta establecida por el usuario, cada dato de información relevante para las tareas requeridas. En base a esto, se debe comprobar que el comando para ejecutar el script estaba formulado correctamente, luego comprobar si el elemento que el argumento del comando estaba solicitando existía dentro de la carpeta ingresada, y finalmente si la carpeta ingresada también existe dentro de los directorios. Este paso se esquematizo en la Fig. 6.

Figura 6. Diagrama sobre toma de Datos simulados

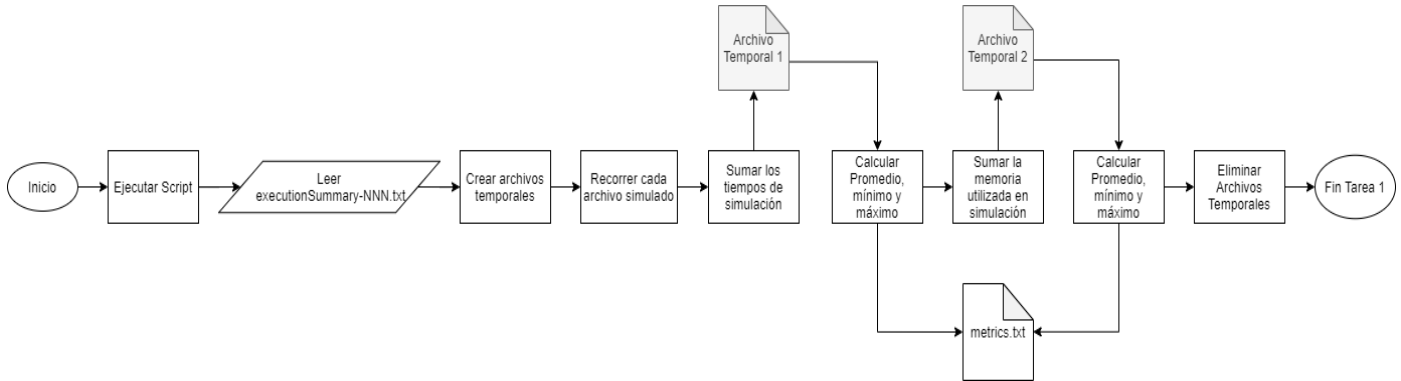


Es posible comprobar que para que se ejecuten las siguientes tareas, se debe comprobar que tanto el argumento ingresado como el directorio a buscar cada archivo, debe existir, si no el programa debe cerrar su ejecución y limitarse a notificar al usuario sobre algún problema y posibles recomendaciones de uso frente a algún error en concreto.

2.3.2. Tarea 1:

Una vez se comprueba la existencia del directorio se buscan los archivos correspondientes, en este caso se debe buscar todos los archivos “executionSummary.txt” que se relacionan con las 11 simulaciones, y una vez realizado esto se propone el siguiente diseño de la solución para el procesamiento y calculo estadístico de los datos de cada archivo de la simulación, expresado en la Fig.7.

Figura 7. Propuesta Diseño solución Tarea 1

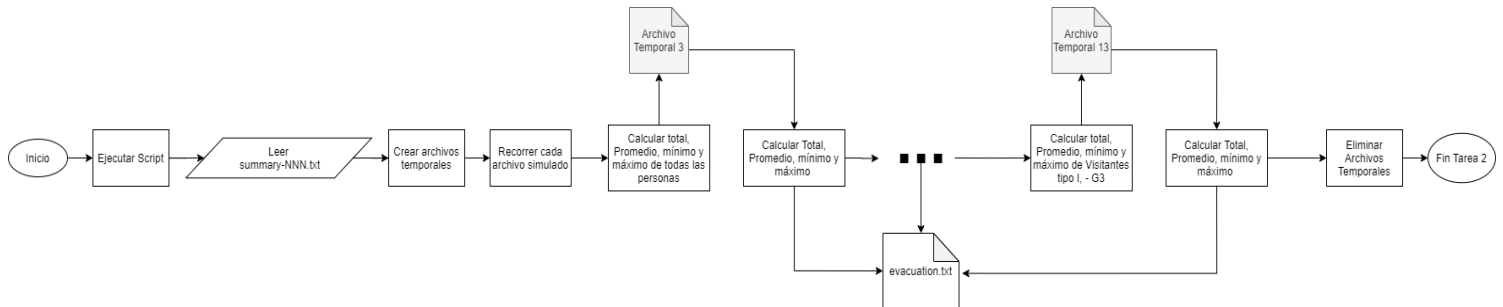


El proceso de ejecución del Script corresponde también al proceso de selección de los archivos. Además, se entiende que se debe almacenar cada dato seleccionado del archivo de entrada, dentro de una variable o archivo temporal, donde en este caso se utilizó la última opción. Tras esto último, se decide calcular el promedio, el mínimo y el máximo de los datos del archivo temporal, para luego ingresar el resultado en el archivo de salida “metrics.txt” y realizar el mismo proceso en cuanto al cálculo de la memoria utilizada por el simulador. Una vez terminado el proceso de cálculo, se procede a eliminar los archivos temporales creados, para finalizar la Tarea 1.

2.3.3. Tarea 2:

Tras completar la Tarea 1 se procede a analizar de la misma manera el inicio de la tarea 2, seleccionando cada archivo “summary-NNN.txt” de cada simulación, aunque este proceso se exige mucho cálculo de estadísticas, por lo que es necesario analizar el diseño más detenidamente en la Fig.8.

Figura 8. Propuesta Diseño de solución Tarea 2



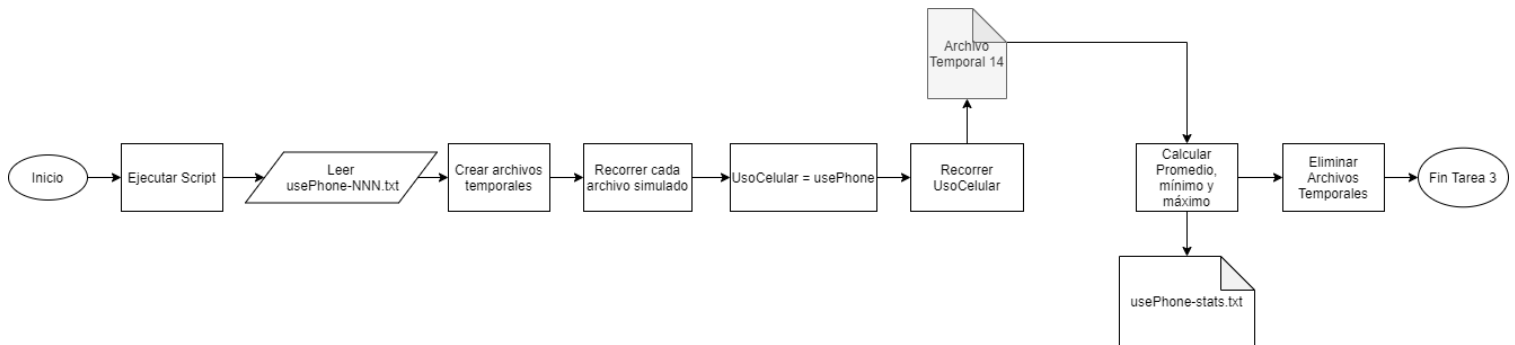
En base al diseño es posible observar que cada vez que se calcule el total, promedio, mínimo y máximo de cada archivo de simulación, se almacenaran estos valores dentro de un archivo temporal, para luego calcular el total completo, el promedio de los promedios, el mínimo de los mínimos y el máximo de los máximos, para que estos datos puedan ser almacenados dentro del archivo de salida “evacuation.txt”. Este proceso de cálculo estadístico entre toma de datos, calculo y escritura en archivo de salida se tomará en cuenta para cada requerimiento solicitado por la tarea, ya que se debe calcular datos estadísticos de todas las personas de la simulación, todos los residentes, todos los visitantes tipo I, solo residentes agrupados por grupo etario y solo visitantes tipo I agrupados también por grupo

etario. Una vez finalizado el proceso de cálculo y escritura se procederá a eliminar los archivos temporales del proceso y se dará por finalizada la Tarea 2.

2.3.4. Tarea 3:

En esta tarea se registra el promedio de uso de teléfonos móviles durante intervalos de tiempo, además de su mínimo y máximo de uso, el diseño de la solución se ve demostrado en la Fig. 9.

Figura 9. Propuesta Diseño de solución Tarea 3

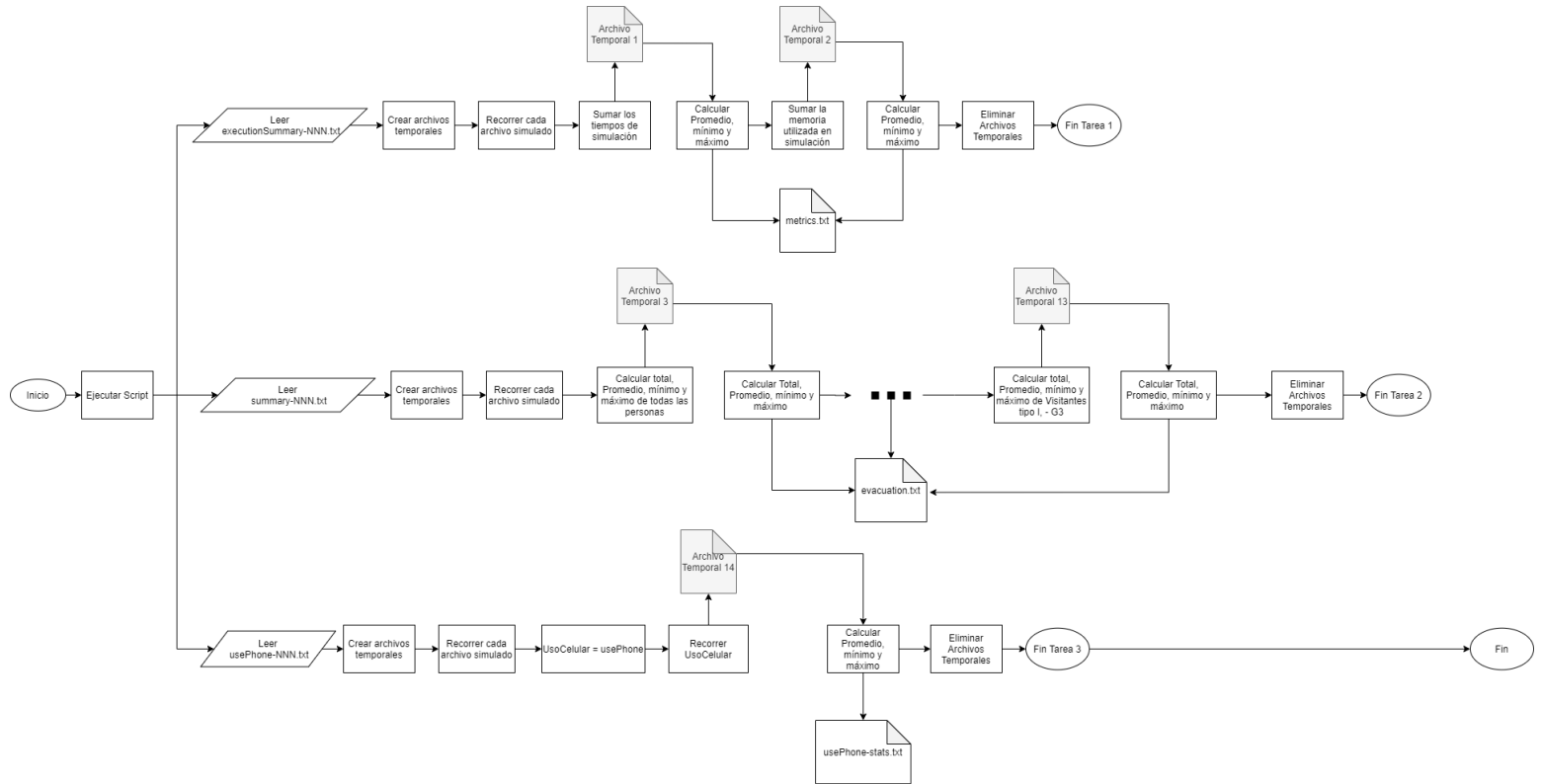


Este procedimiento es más sencillo, ya que, una vez terminado de recorrer cada archivo simulado, se procederá a asignar el uso de teléfonos dentro de una variable “UsoCelular”, el cual servirá para recorrer y almacenar dentro de un archivo temporal de manera que calcule el promedio, mínimo y máximo para ser almacenados en el archivo de salida “usePhone-stats.txt”, eliminando finalmente el archivo temporal y finalizando la tarea 3.

2.4. Diseño Final:

Una vez diseñado e implementado la resolución de cada tarea, es necesario resumir la implementación del Script uniendo la resolución de todas las tareas dentro de un solo archivo de comandos. Como se especificó con anterioridad, todas estas tareas deben ser ejecutadas por el Script “stats.sh” el cual posee el siguiente diseño en vista de las tareas que realiza para solucionar el taller, todo esto se encuentra expresado en la Fig. 10.

Figura 10. Propuesta Diseño General de solución Final



Se puede observar que al ejecutar el Script, se selecciona cada archivo de las simulaciones encontradas en los directorios de estos, donde al tomar cada archivo buscado, se realiza el proceso de calcular estadísticas y datos para almacenarlos dentro de archivos temporales para luego tomar esta información a la hora de realizar los datos estadísticos finales y proceder a registrarlos en el archivo de salida establecido para cada tarea, finalizando cada una de las tareas para terminar todo el script al completar la Tarea 3.

3. Referencias.

- [1] Linux Command Line, William Shotts, Fifth Internet Edition, 2008-2019, pp.02-80.