

ANÁLISIS DE ORDENACIÓN Y SELECCIÓN DE GRUMOS

Una vez que se ha procesado el documento y se han creado de manera correcta los grupos, es paso de ordenarlos y seleccionar los necesarios para cumplir con el porcentaje que el usuario desea.

Previamente, durante la ejecución de la función `pedirDatos()`, se le pidió al usuario que introdujera el porcentaje de mayor tamaño de un grupo. En este caso, lo que se busca es realizar una serie de conexiones entre unos usuarios de los grupos más grandes para conseguir que exista un grupo que englobe al porcentaje de usuarios deseado.

Tras haber finalizado la función `crearGrupos()`, se ejecuta la función `ordenarGrupos()`. Esta función será la encargada de ordenar los grupos de mayor a menor número de usuarios. Para ello, recorre el listado de grupos y recoge el primero y el siguiente (es decir, durante la primera iteración cogerá el grupo 1 y el 2, en la segunda el 2 y el 3, en la tercera el 3 y el 4, y así sucesivamente), y va comparándolos hasta que ya no exista un grupo siguiente (final del listado). La comprobación que realiza es si el segundo grupo recogido es mayor en número de usuarios que el primero. Si así es, se almacena ese grupo en un `ArrayList` temporal (para evitar pérdida de datos), se elimina el grupo de la lista y se añade, con la función `add()`, el `ArrayList` temporal en la posición del anterior grupo.

La función `add()` permite dos parámetros, el primero el índice en el que se quiere guardar un objeto, y el segundo el objeto. Si se agrega un objeto en una posición en la que ya existía otro, éste será desplazado a la posición siguiente. Por ello, al finalizar la ejecución, los grupos estarán ordenados de mayor a menor número de usuarios.

Finalizada la ordenación, queda seleccionarlos. Para ello, se recorre la lista de grupos (ya ordenada) y se van sumando los usuarios del grupo actual y el siguiente hasta que se alcance el porcentaje deseado por el usuario. El porcentaje se calcula en cada iteración para comprobar que sea correcto. Una vez que el porcentaje calculado es superior o igual al porcentaje deseado, se corta la ejecución del bucle y se imprime por pantalla los resultados (número de grupos mayores que han de unirse y los usuarios que se deben unir).

Los usuarios a unir siguen un orden de i , es decir, que si el primer grupo es el grupo 1 (i) y el segundo es el grupo 2 ($i+1$), será el usuario 1 del grupo 1 el que se una con el usuario 2 del grupo 2. Si en algún momento no existiera el usuario necesario (por ejemplo, si hay 5 usuarios y se necesita unir con el usuario 6), volvería a seleccionar el primero.

Finalmente, ejecuta la función `salvarConexiones`, que almacenará en el archivo "extra.txt", en cada línea, las conexiones necesarias para que, ejecutando por segunda vez el programa y seleccionando un archivo de conexiones extensas, se compruebe que las conexiones se realizan correctamente y que el grupo mayor es el deseado.

A continuación, captura del resultado del proceso, creando un grupo que englobe al 90% de los usuarios:

```

Output - CaraLibro (run) X
run:
ANÁLISIS DE CARALIBRO
-----
Fichero principal: test20000.txt
Lectura fichero: 0,01800 seg.
Fichero de nuevas conexiones (pulse enter si no existe):
20000 usuarios, 19919 conexiones
Porcentaje tamaño mayor grupo: 90
Creación lista usuarios: 0,93100 seg.
Creación lista grupos: 3,39700 seg.
Existen 81 grupos.
Se deben unir los 5 mayores
#1: 4296 usuarios 21.0 %
#2: 4290 usuarios 21.0 %
#3: 4294 usuarios 21.0 %
#4: 4292 usuarios 21.0 %
#5: 2652 usuarios 13.0 %
Nuevas relaciones de amistad (salvadas en extra.txt)
16252944 <-> 48103433
48103433 <-> 79908463
79908463 <-> 85370320
85370320 <-> 75387301
Ordenación y selección de grupos: 0,00000 seg.
BUILD SUCCESSFUL (total time: 17 seconds)

```

```

Output - CaraLibro (run) X
run:
ANÁLISIS DE CARALIBRO
-----
Fichero principal: test20000.txt
Lectura fichero: 0,01500 seg.
Fichero de nuevas conexiones (pulse enter si no existe): extra.txt
20000 usuarios, 19923 conexiones
Porcentaje tamaño mayor grupo: 90
Creación lista usuarios: 1,01300 seg.
Creación lista grupos: 3,74400 seg.
Existen 77 grupos.
El mayor grupo contiene 19824 usuarios (99.0)%
No son necesarias nuevas relaciones de amistad.
Ordenación y selección de grupos: 0,00000 seg.
BUILD SUCCESSFUL (total time: 16 seconds)

```