**ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORES**

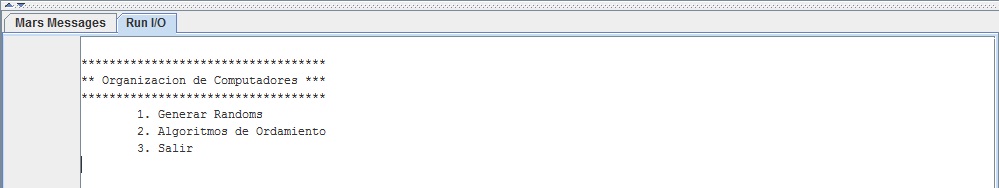
**WRITE UP - PROYECTO PRIMER PARCIAL**

Nuestro proyecto del primer parcial de la materia de Organización y Arquitectura de Computadores constaba de 4 literales que resumiremos en pocas palabras. Básicamente lo que se pedía en el proyecto era crear un programa en el que se pudiera realizar las siguientes acciones: Generar Archivo de Números Aleatorios, y Ordenar los Grupos de Números Aleatorios del Archivo utilizando algoritmos de ordenamiento. Adicional a esto se pidió también que los tiempos que se tomó en realizar el ordenamiento de los grupos sean interpretados mediante gráficas, además todas las acciones mencionadas anteriormente debían poder ser escogidas por el usuario mediante un menú.

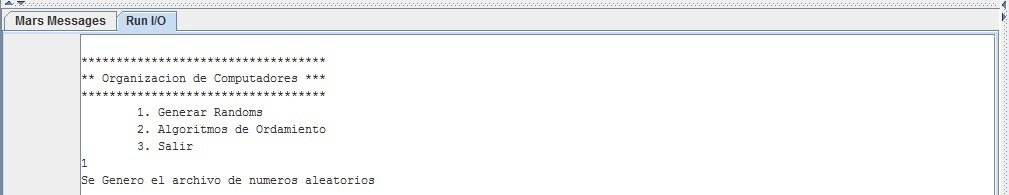
Cada una de las partes del proyecto son detallas a continuación:

**Parte 1: Generar Archivo de Números Aleatorios**

La primera era que se pueda generar un archivo de nombre “aleatorios.txt” con 10 grupos de números aleatorios que estuvieran en un rango entre el 0 y el 1000. En el menú de nuestro programa dicha acción está representada por la opción: *Generar Randoms*.

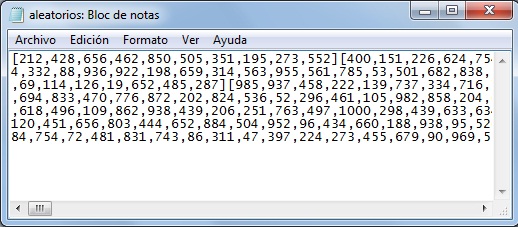


**1.a) Menú Principal del Programa.**

****

**1.b) Menú Principal luego de seleccionar la opción 1.**

El formato que escogimos para nuestro archivo que contiene los grupos de números aleatorios separamos los grupos por corchetes, y los números de cada grupo los separamos por comas. A continuación se muestra el formato: [num1\_grupo1,num2\_grupo1,...,num10\_grupo1][num1\_grupo2,num2\_grupo2,….,num20\_grupo2]……………….….[num1\_grupo10,num2\_grupo10,……,num400\_grupo10].

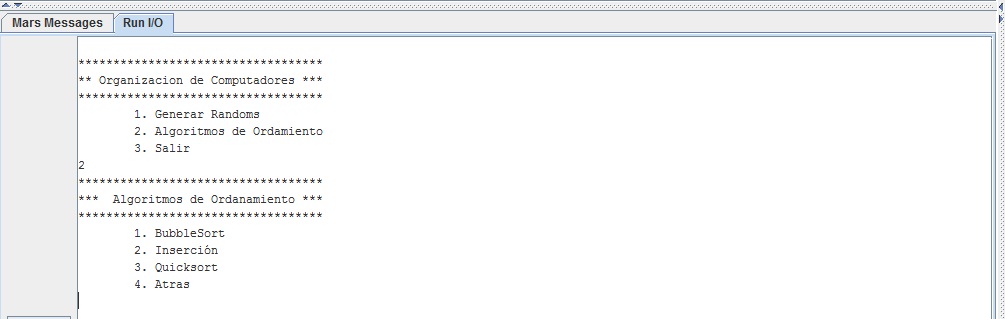
****

**1.c) Archivo “aleatorios.txt” generado.**

**Parte 2: Ordenar los Grupos de Números Aleatorios del Archivo**

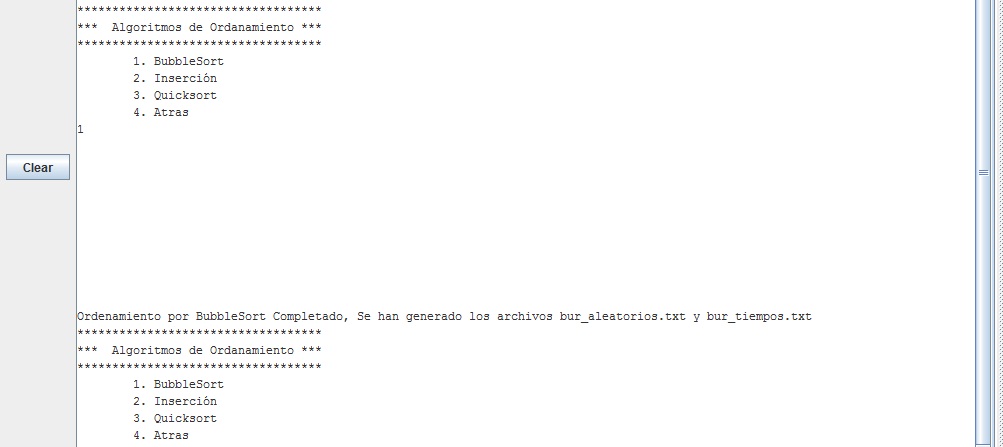
La segunda parte consistía en que el archivo generado en la primera parte debería poder ser leído por el programa, para posteriormente poder ordenar cada uno de los 10 grupos de números aleatorios, esto utilizando un algoritmo de ordenamiento seleccionado por el usuario. Los algoritmos que se pidió que estuvieran disponibles para ser seleccionados por el usuario son los siguientes: Bubble Sort, Insertion Sort, y Quick Sort.

Una vez ordenados los grupos de números se debían generar dos archivos: uno que contenga los grupos ya ordenados, y otro que contenga los tiempos que se tomó hacer el ordenamiento seleccionado por el usuario.

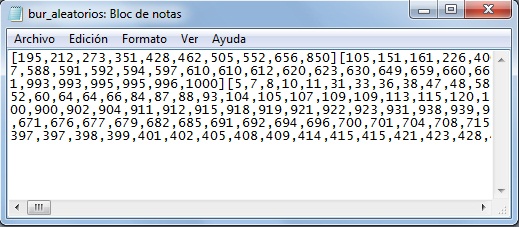


**2.a) Menú Principal luego de seleccionar la opción 2.**

Luego de seleccionar en el menú principal la opción 2, nos aparecerá un nuevo menú donde podremos seleccionar el ordenamiento que queramos.

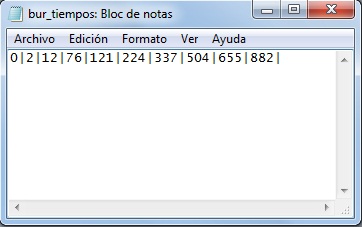


**2.b) Menú Algoritmos de Ordenamiento luego de seleccionar la opción 1.**

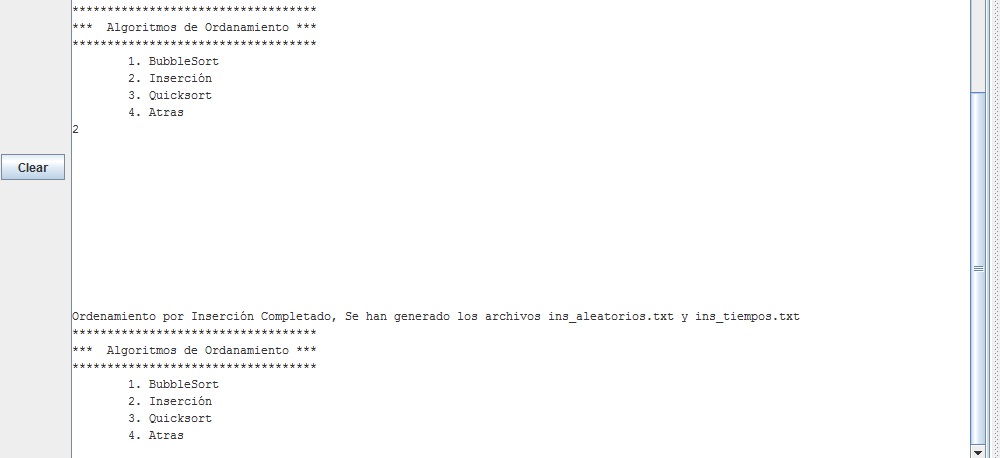
****

**2.c) Archivo “bur\_aleatorios.txt” generado.**

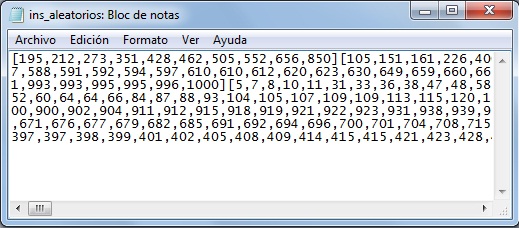
En el formato que escogimos para el archivo que contienen los tiempos separamos el tiempo de cada grupo con una barra vertical.

****

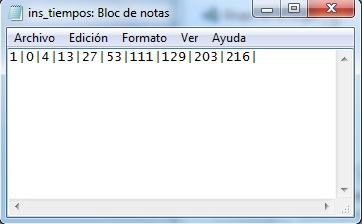
**2.d) Archivo “bur\_tiempos.txt” generado.**

****

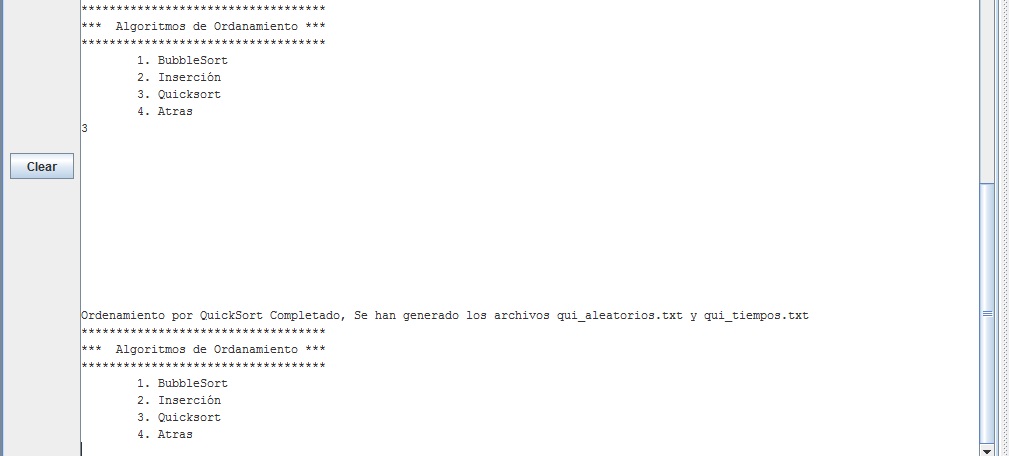
**2.e) Menú Algoritmos de Ordenamiento luego de seleccionar la opción 2.**

****

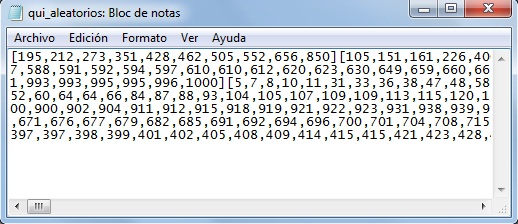
**2.f) Archivo “ins\_aleatorios.txt” generado.**

****

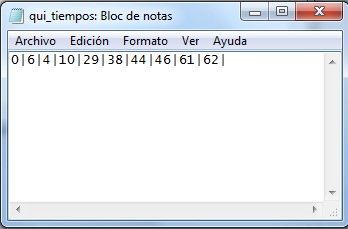
**2.g) Archivo “ins\_tiempos.txt” generado.**

****

**2.h) Menú Algoritmos de Ordenamiento luego de seleccionar la opción 3.**

****

**2.i) Archivo “qui\_aleatorios.txt” generado.**

****

**2.j) Archivo “qui\_tiempos.txt” generado.**

**Parte 3: Interpretar los tiempos de los ordenamientos**

Como parte final del proyecto se pidió que los datos de tiempo guardados en los distintos archivos, sean interpretados mediante gráficas, utilizando algún software.