

## Descripción de los archivos presentados

A continuación se describen varias consideraciones a tener en cuenta a la hora de ejecutar el código, leer los resultados, establecer los parámetros de configuración y datos de entrada.

### Organización de los archivos

Entre los archivos que se entregan existe código fuente (.java), código ejecutable (.class) y archivos de datos y configuración (.txt). Todos los archivos tienen codificación UTF-8, por tanto, debe asegurarse que al editarlos o visualizarlos, lo haga con un editor o visor que soporte UTF-8.

En cuanto al código, el mismo está organizado en paquetes. A continuación se da una descripción de cada uno de los paquetes y sus respectivas clases. Mayor información se encontrará en los comentarios del código fuente.

Paquete	Clases/Interfaces	Descripción
Enrut	Algoritmo, Demanda, Solucion	Clases relacionadas a ambos algoritmos.
enrut.ag	Cromosoma, Poblacion	Clases relacionadas exclusivamente al algoritmo basado en AG.
enrut.ag.oper	OperadorCruce, OperadorMutacion, OperadorSeleccion	Interfaces de los operadores relacionados al algoritmo basado en AG.
enrut.ag.impl		No contiene ninguna clase pero sí paquetes de las implementaciones de los operadores.
enrut.ag.impl.cruce	CruceUniforme	Clase que implementa la interfaz de cruce.
enrut.ag.impl.mutacion	MutacionGenes	Clase que implementa la interfaz de mutación.
enrut.ag.impl.seleccion	TorneoBinario	Clase que implementa la interfaz de selección.
enrut.grafo	Arista, Camino, ConstructorCaminos, ConstructorGrafo, GeneradorCaminos, Grafo, GrupoCaminos, PrintTo, Vertice	Clases relacionadas con el grafo y la generación de rutas del grafo.
enrut.pso	Enjambre, Particula	Clases relacionadas exclusivamente al algoritmo basado en PSO.
enrut.pso.oper	Movimiento	Interfaz de la operación de movimiento de las partículas
enrut.pso.oper.impl	MovimientoTradicional	Clase que implementa la interfaz de movimiento de las partículas
enrut.test	AlgoritmoAG, AlgoritmoPSO, GenerarCaminos, EjecutarAG, EjecutarPSO, Principal	Las clases EjecutarXYZ son las que ejecutan cada uno de los algoritmos. El algoritmo principal está en la clase Principal. La clase GenerarCaminos genera los caminos para un determinado grafo.
enrut.utils	Config, CSVWriter, Escritor, ImpresionSalida, FormatDir, Lector, LectorTeclado	Clases de utilería, normalmente utilizadas para entrada/salida.

### Compilación del código

Ya se entrega el código compilado en código ejecutable (.class). La manera más fácil de recompilar alguna de las clases sería abriendo el proyecto con Eclipse y compilarlo desde allí, ya que la naturaleza de la organización de las clases en paquetes hace muy tediosa su compilación por línea de comandos.

De todas maneras, la compilación de las clases de cada paquete podría realizarse yendo hasta el directorio donde se encuentra el paquete **enrut** y debería hacer lo siguiente:

```
javac -encoding utf8 -d <directorio_destino> enrut\test\GenerarCaminos.java
javac -encoding utf8 -d <directorio_destino> enrut\test\Ejecutar*.java
```

La variable `<directorio_destino>` es el directorio en el que se guardarán los archivos .class generados. La versión del compilador utilizado para compilar debe ser 1.6.0.

Existen tres clases ejecutables dentro del proyecto, las cuales son: **enrut.test.GenerarCaminos**, **enrut.test.EjecutarAG**, **enrut.test.EjecutarPSO**, las cuales ejecutan, respectivamente, la generación de rutas de un grafo, el algoritmo basado en AG y el algoritmo basado en PSO, según los archivos de datos y configuración. Para ejecutar cualquiera de las clases ejecutables desde línea de comandos, debería realizar ir hasta el directorio donde se encuentra el paquete **enrut** y debería hacer lo siguiente:

```
java enrut/test/<clase> <directorio_configuracion>
```

La variable `<clase>` puede tomar cualquier de los tres valores: EjecutarAG, EjecutarPSO o GenerarCaminos. La variable `<directorio_configuracion>` se refiere al directorio en el que se encuentran los archivos de datos y de configuración.

### Archivos de entrada (datos de entrada) y de configuración

Para la ejecución de los tres archivos ejecutables que se mencionaron anteriormente, se necesitan archivos de datos de entrada. A continuación se detalla la necesidad de cada uno.

Ejecutable	Archivo	Tipo	Observación
enrut.test.GenerarCaminos	grafo.txt	Datos	Archivo que contiene la definición del grafo. En el mismo debe aparecer en la primera y segunda fila, la cantidad de nodos y aristas, respectivamente. En las subsecuentes líneas está la definición de cada una de las aristas: origen, destino, capacidad, costo, retardo.
enrut.test.EjecutarAG enrut.test.EjecutarPSO			Archivo que contiene las demandas solicitadas, sobre las que trabajarán los algoritmos. En el mismo debe aparecer en la primera línea la cantidad de demandas. En las subsecuentes líneas debe aparecer cada una de las demandas: origen, destino, ancho de banda.
		Configuración	Archivo que contiene los datos de configuración para la ejecución de los algoritmos. Puede tener líneas en blanco y comentarios de línea (comenzando con #). Más adelante se describe cada una de las variables de configuración.

Como parámetro de estos ejecutables debe pasárseles el directorio donde se encuentran estos archivos de datos o de configuración, los cuales deben encontrarse en el mismo directorio. Toda la salida (archivos) de los ejecutables se realiza en este mismo directorio.

## Archivo de configuración

El archivo de configuración **config.txt**, cuenta con varias variables cada una de las cuales debe estar presente en la forma de par clave-valor, de la siguiente manera: “clave = valor”. Entre la *clave* y el signo “=”, y entre éste último y el *valor*, puede haber espacios en blanco. Las variables son las siguientes:

Variable	Descripción
MAX_TIEMPO	Máximo tiempo que durará cada ejecución del algoritmo, en segundos.
MAX_CAMINOS	Máxima cantidad de rutas a cargar
TAM_POBLACION_AG	Tamaño de la población del algoritmo basado en AG
TAM_POBLACION_PSO	Tamaño de la población del algoritmo basado en PSO
CANT_EJECUCIONES	Cantidad de ejecuciones a realizar
INTERVALO_MUESTRA	Intervalo de muestreo de los algoritmos, en milisegundos
PROB_MUTACION_AG	Probabilidad de mutación del algoritmo basado en AG
PORC_REINICIO_AG	Porcentaje de reinicialización del algoritmo basado en AG
PORC_REINICIO_PSO	Porcentaje de reinicialización del algoritmo basado en PSO
TABLA_RUTAS	Subdirectorio dentro del directorio de configuración, dentro del cual se encuentran los archivos de rutas para las demandas.

Las variables pueden aparecer en cualquier orden, con la única restricción de que MAX\_CAMINOS debe aparecer antes de TABLA\_RUTAS.

### Archivos de resultados

Los archivos generados por los ejecutables **enrut.test.EjecutarAG** y **enrut.test.EjecutarPSO**, son archivos con el formato CSV (comma separated values), los cuales pueden ser abiertos en MS Excel o OpenOffice Calc para su edición y generación de estadísticas y gráficos. El patrón de nombre de los archivos es:

```
historico<numero_ejecucion>-<algoritmo>.csv
```

La variable `<numero_ejecucion>` es el número de ejecución que se realiza, ya que pueden realizarse más de una (ver variables de configuración).

La variable `<algoritmo>` indica el algoritmo, pudiendo contener "AlgoritmoAG" o "AlgoritmoPSO".

### Archivos de rutas

Los archivos generados por el ejecutable **enrut.test.GenerarCaminos**, son archivos que contienen las rutas de un grafo. El patrón de nombre que tienen los archivos es:

```
caminos-<origen>-<destino>.txt
```

En la primera y segunda línea del archivo aparecen, respectivamente, el origen y destino del camino. A partir de la tercera línea aparecen las rutas generadas. El formato de cada ruta es el siguiente:

```
<longitud>-<costo>-<capacidad_minima>-<aristas>
```

Donde `<aristas>` es la lista de aristas de la ruta correspondiente y puede contener uno o más de lo siguiente, separado por guiones:

```
<origen>:<destino>:<capacidad>:<costo>
```

Las rutas para el grafo NSFNet ya se encuentran generadas en la carpeta de *rutas* del directorio de datos del código fuente. La generación de rutas para este grafo dura aproximadamente 3 segundos. No se realizaron pruebas con otros grafos.