

GEOACTA 40(1): 1-9 (2015)
© 2015 Asociación Argentina de Geofísicos y Geodestas

ISSN 1852-7744

APLICACIONES INFORMÁTICAS PARA GENERAR Y ARREGLAR ARCHIVOS DE NAVEGACIÓN Y REGISTROS SÍSMICOS (SEGY)

INFORMATIC APPLICATIONS TO GENERATE AND FIX NAVIGATION FILES AND SEISMIC RECORDS (SEGY)

Esteban, F.D.¹, Esteban, A.H.²

¹ CONICET-Instituto de Geociencias Básicas, Ambientales y Aplicadas (IGeBA), Depto. de Cs. Geológicas, Fac. de Cs. Exactas y Naturales, Universidad de Buenos Aires, Argentina.

² CONICET-Instituto de Geociencias Básicas, Ambientales y Aplicadas (IGeBA), Departamento de Cs. Geológicas, Fac. de Cs. Exactas y Naturales, Universidad de Buenos Aires, Argentina.

E-mail: esteban@gl.fcen.uba.ar

RESUMEN

Se presentan cinco programas simples escritos en lenguaje JAVA con dos fines. El primer grupo de programas permite generar archivos de navegación de forma efectiva y rápida para utilizar en los casos en que los archivos Segy no tengan incorporados estos datos. Estos programas crean archivos de navegación, compatibles con el Kingdom Suite y el Generic Mapping Tools (GMT) a partir de los datos de localización de las líneas sísmicas y de los respectivos números de puntos de energización. También calculan datos estadísticos (longitud total de la línea, equidistancia entre puntos de explosión) a partir de los archivos de navegación generados. Esta información estadística permite determinar la calidad de las fuentes de datos de navegación consultadas (y compararlas si hay más de una). El segundo conjunto de programas permite arreglar dos tipos de errores identificados en los archivos Segy. El primero se caracteriza por un desfase en un sector del perfil sísmico que se atribuyó a la falta de datos en la traza ubicada justo antes del desfase. En el segundo tipo de error los archivos son totalmente ilegibles y se asumió que el inicio del perfil sísmico se localizaba en un posición intermedia del archivo.

Palabras Claves: Sísmica, Navegación, JAVA, Archivos Segy, software científico.

ABSTRACT

We present five programs written in the JAVA language with two purposes in mind. The first allows one to elaborate navigation files in an effective and rapid way to be used when the Segy files lack navigation data. These programs create navigation files compatible with the Kingdom Suite and the Generic Mapping Tools (GMT) from a database of the seismic line locations and their respective shot point numbers. It also calculates statistical data (total length, shot point equidistance) for each file generated, which helps determine the quality of the navigation data sources and compare among them for multiple files. The second set of programs allows one to fix two types of errors identified in the Segy files. The first is characterized by an offset within the seismic section that was attributed to lack of data in the trace just previous the offset. For second error the files were totally illegible and it is assumed that the beginning of the seismic section was somewhere in the middle of the file.

Keywords: Seismic, Navigation, JAVA, Segy files, Scientific software.

INTRODUCCIÓN

Los Segy son los archivos estándar para el almacenamiento de datos sísmicos (Noris y Faichney, 2002). Estos archivos, además, contienen información sobre las características del relevamiento, incluyendo números de puntos de energización (o *shot point*; SP), de navegación, y otros datos que permiten su correcta utilización por parte de programas de interpretación sísmica (ej. Kingdom Suite, SeiSee).

En base a la experiencia de la integración y conformación de base de datos sísmicos, se ha visto que en ocasiones los archivos Segy carecen de datos de navegación o de los números de SP incorporados. En estos

casos es necesario, primero conformar una base de datos de navegación (en ocasiones se consulta más de una fuente), y otra con los números SP de cada línea, y luego generar un archivo de navegación para cada archivo Segy (con número de SP y sus respectivos datos de longitud y latitud). La generación artesanal de estos archivos es un trabajo que requiere unos minutos por cada uno, y constituye una tarea sencilla pero reiterativa y aburrida, que en ocasiones puede introducir errores aleatorios (y por lo tanto difícil de detectar) dentro de los archivos. Para los casos donde es necesario crear una gran cantidad de archivos de navegación, y sumado a las posibles versiones que se pueden crear para cada uno (según la combinación de datos navegación y números de SP provenientes de distintas fuentes) y a la posible corrección de los archivos (debido a los errores aleatorios), la tarea artesanal termina constituyendo un trabajo que consume una gran cantidad de tiempo.

Por otra parte, los archivos Segy pueden tener defectos en su estructura que impiden su correcta visualización. El primero se caracteriza por un desfase en un sector del perfil sísmico, mientras que el segundo tipo de error los archivos son totalmente ilegibles. En ocasiones, es importante poder reparar estos archivos Segy dada su ubicación geográfica estratégica, o de escasa cobertura sísmica (ej. Esteban, 2014).

En este trabajo se discuten 5 programas desarrollados con dos objetivos: creación de archivos de navegación para la integración de una base de datos sísmica de manera rápida y efectiva, y la refacción de dos tipos de errores en archivos Segy.

PROGRAMAS DESARROLLADOS

Los programas se escribieron con el lenguaje de programación JAVA, bajo el entorno de desarrollo Eclipse (versión 3.6). Se desarrollaron para dos propósitos particulares. Los programas Crear_Input_JAVA, NAV_JAVA y BaseData_Java tienen como objetivo la creación de archivos de navegación para cada una de las líneas sísmicas. Los programas son compatibles tanto por el Kingdom Suite (programa para interpretación sísmica) como para el Generic Mapping Tools (GMT; Wessel et al. 2013). En cambio, los programas Relocate_EBCIDC_Header y fileByteFiller permitieron la corrección de archivos Segy.

La ejecución de los programas se realiza con líneas de comando a partir de archivos de texto plano (ej. bloc de notas o gedit). En estos archivos cada línea constituye una orden que se ejecutan en serie, y están conformadas por el nombre de programa y los respectivos argumentos requeridos. En un mismo archivo se pueden incluir numerosas órdenes incluso de programas distintos.

Los mismos se pueden ejecutar en cualquier sistema operativo, solo requieren la instalación del programa JAVA (versión 1.4 o posterior).

FUNCIONAMIENTO DE LOS PROGRAMAS

Crear_Input_JAVA

El programa Crear_Input_JAVA tiene como objetivo crear un tipo de archivo de texto plano (tipo ASCII) el cual luego se utilizará para la generación de archivos de navegación o para graficar directamente con el programa GMT. El programa requiere de dos argumentos:

```
java -jar Crear_Input_JAVA.jar "Tesis_BB.txt" "Tesis_BB"
```

nombre del archivo de entrada (Tesis_BB.txt, en el ejemplo anterior) y nombre de la carpeta de salida (Tesis_BB). El archivo de entrada requiere una estructura particular. Para su elaboración se utiliza el programa Global Mapper (versión 13) el cual presenta la ventaja de una gran versatilidad para leer una importante variedad de archivos vectoriales e imágenes ráster. Una vez cargados todos los datos de navegación (de una o varias líneas) en el Global Mapper, se exportan como archivo vectorial del tipo archivo de texto plano ASCII (*Simple ASCII Text File*) con las opciones que se pueden ver en la Figura 1A. La separación de las coordenadas por tabulaciones (tab) permite que el programa Crear_Input_JAVA reconozca las columnas, y la separación con una línea en blanco (*blank line*) entre rasgos permite la identificación de cada una de las línea (Figura 1B). La inclusión de los atributos, antes de las coordenadas, es importante porque incluye el nombre de la línea (*Name=*), que luego el programa utilizará para el nombrar al archivo generado.

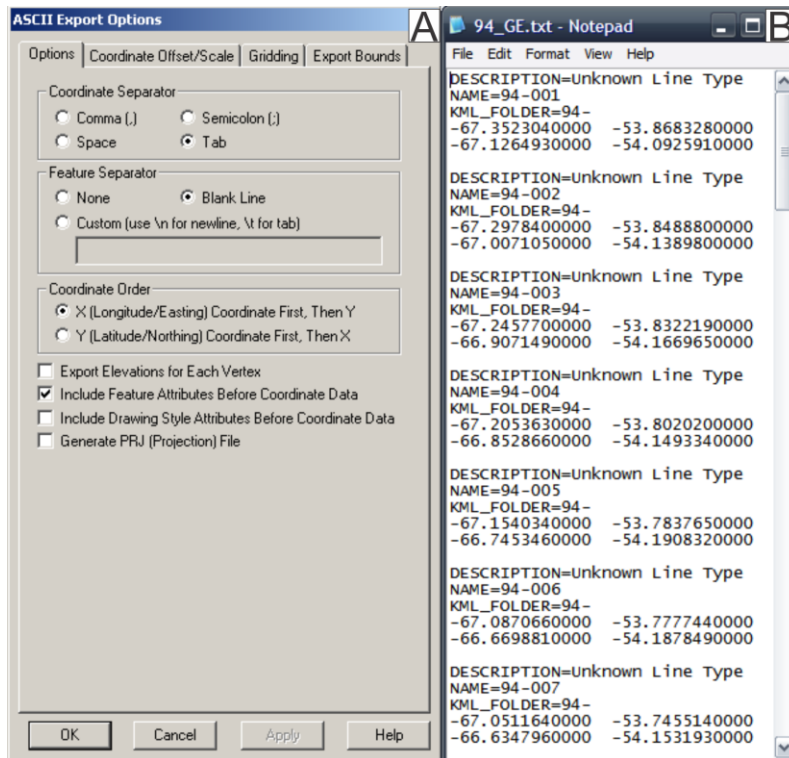


Figura 1. A) Imagen del programa Global Mapper (v. 13) con las opciones para la exportación de archivos de texto plano tipo ASCII para utilizar el programa de línea de comando Crear_Input_JAVA. B) Imagen del archivo 94_GE.txt exportado según los parámetros definidos en A.

Figure 1. A) Global Mapper (v. 13) image with the export options for plain text ASCII type to use as input for the line-command Crear_Input_JAVA software. B) Image of the file 94_GE.txt exported following the parameters defined in A.

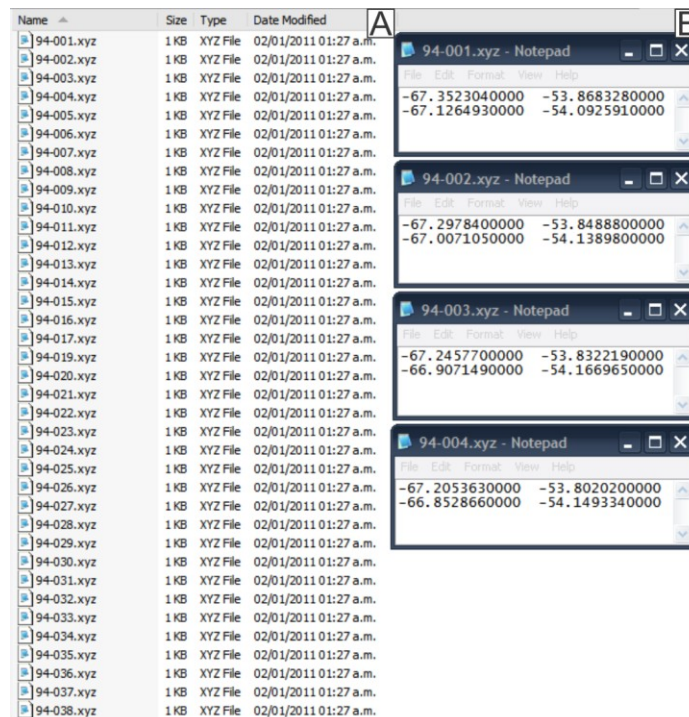


Figura 2. A) Captura de pantalla del directorio con los archivos generados por el programa Crear_Input_JAVA. Nótese que todos los archivos (38) fueron creados en el mismo minuto. B) Ejemplos de los archivos generados que luego se utilizarán como archivos de entrada para el programa NAV_JAVA.

Figure 2. A) Screen capture of the folder with the files generated by Crear_Input_JAVA software. To be noticed that all the files (38) were created in the same minute. B) Examples of the files generated that then will be used as input files for the NAV_JAVA software.

Crear_Input_JAVA.jar genera un único archivo para cada línea cuyo nombre corresponde al texto que aparece en la 2ª fila después del signo “=” (Figura 1B). Los archivos generados son de tipo texto plano tipo ASCII (con extensión xyz), y contienen únicamente dos columnas con las coordenadas geográficas (longitud y latitud; Figura 2). En el caso de que haya más de un archivo con el mismo nombre, el programa agrega las nuevas coordenadas al primer archivo creado. Los archivos generados pueden ser graficados por el GMT.

El programa puede manejar sin complicaciones navegaciones de distinta complejidad. Es decir, donde la navegación sigue líneas rectas (conformadas por dos puntos, Figura 2) o presentan cambios de dirección (conformada por muchos puntos; Figura 3). Esto resulta importante por ejemplo en los relevamientos de sísmica monocal del Atlántico sudoccidental (ver Esteban, 2014).

#Long	Lat	SP	Línea	X (UTM-20)	Y (UTM-20)	Equidist (m)= 24.99	Length (m)= 89425.93
SP 0=	1101	SP f=	4679	Fuente NAV=	GE	Fuente SPs=	MI
Archivo NAV= INPUT\YCM_GE\YCM-82.xyz							
-62.465354	-53.546235	1101	YCM-82	535425.666	4066829.597		
-62.468192	-53.547334	1110	YCM-82	535236.708	4066708.741		
-62.47139	-53.548517	1120	YCM-82	535023.839	4066578.708		
-62.474604	-53.549699	1130	YCM-82	534809.923	4066448.785		
-62.477818	-53.550901	1140	YCM-82	534596.002	4066316.628		
-62.480972	-53.552119	1150	YCM-82	534386.055	4066182.652		
-62.484102	-53.553381	1160	YCM-82	534177.674	4066043.76		
-62.48721	-53.554653	1170	YCM-82	533970.756	4065903.736		
-62.490349	-53.55589	1180	YCM-82	533761.823	4065767.611		
-62.493533	-53.557108	1190	YCM-82	533549.937	4065633.612		
-62.496688	-53.558325	1200	YCM-82	533339.985	4065499.701		
-62.499842	-53.559569	1210	YCM-82	533130.09	4065362.777		
-62.502989	-53.560804	1220	YCM-82	532920.678	4065226.841		
-62.506144	-53.562042	1230	YCM-82	532710.746	4065090.566		
-62.509284	-53.563277	1240	YCM-82	532501.821	4064954.608		
-62.512439	-53.564493	1250	YCM-82	532291.93	4064820.762		
-62.515561	-53.565719	1260	YCM-82	532080.984	4064685.801		
-62.518781	-53.566938	1270	YCM-82	531870.054	4064551.61		
-62.521974	-53.568155	1280	YCM-82	531657.682	4064417.641		
-62.525161	-53.569354	1290	YCM-82	531445.732	4064285.663		
-62.528317	-53.57057	1300	YCM-82	531235.834	4064151.77		
-62.531488	-53.571787	1310	YCM-82	531024.955	4064017.764		
-62.534645	-53.573023	1320	YCM-82	530815	4063881.628		
-62.537817	-53.57424	1330	YCM-82	530604.078	4063747.603		
-62.540996	-53.575465	1340	YCM-82	530392.699	4063612.681		
-62.544153	-53.57669	1350	YCM-82	530182.789	4063477.741		
-62.547296	-53.577924	1360	YCM-82	529973.812	4063341.784		
-62.550438	-53.57916	1370	YCM-82	529764.911	4063205.595		
-62.553595	-53.580394	1380	YCM-82	529555.031	4063069.626		
-62.556768	-53.581618	1390	YCM-82	529344.11	4062934.766		

Figura 3. Ejemplo de archivo de navegación generado con el programa NAV_JAVA para la línea YCM-82. La primera fila (graficada en 2 filas para poder visualizarla) corresponde al encabezado del archivo (ver descripción en texto). Las primeras 6 columnas conforman la primera parte del encabezado y describen la información del cuerpo del archivo (resto de las filas). La segunda parte del encabezado (resto de las columnas) incluye los datos y sus respectivas fuentes utilizados para los cálculos realizados por NAV_JAVA (ver texto).

Figure 3. Example of the navigation file generated by the NAV_JAVA program for the line YCM-82. The first row (split in 2 in the figure) corresponds to the file header (see description in text). The first 6 columns conform the header first part and contain the information describe in the body of the files (rest of the rows). The second part of the header (rest of the columns) includes the data and the respective sources used for calculations by NAV_JAVA (see text).

NAV_JAVA

El programa NAV_JAVA conforma el segundo paso para la generación de los archivos de navegación. El mismo consiste en asignarle a cada archivo xyz, generado previamente, el número de SP para el inicio y el final de la línea.

NAV_JAVA requiere de 8 argumentos:

```
java -jar JAR\NAV_JAVA.jar "INPUT\94_MI\94-02.xyz" "150" "1650" "94-02" "94_MI" "MI" "MI" "-n"
```

Los más importantes son los tres primeros que describen el nombre del archivo con la navegación (creado previamente por Crear_Input_JAVA), el número de SP inicial (SP 0) asignado a un extremo de la línea, el n° de SP final (SP f) asignado al otro extremo. El último argumento sirve para los casos en que se observa que la posición del perfil sísmico quedó invertida con respecto a la batimetría, con rasgos geológicos evidentes (ej. posición del basamento) o al cruce con otras líneas sísmicas. Este argumento define si la asignación de los números de SP es normal (“-n”) o inverso (“-i”). El 4° argumento corresponde al nombre de la línea (y del archivo), mientras que el 5° define el nombre del directorio donde se guardan los archivos generados.

Los demás argumentos brindan información sobre las fuentes consultadas. El 6° y 7° argumento describen la fuente de los datos sobre el número de SP, y los datos de navegación, respectivamente.

Con los argumentos dados, el programa realiza los siguientes pasos:

1. Conversión de las coordenadas geográficas a coordenadas planas (UTM-20) para luego poder hacer cálculos matemáticos con las coordenadas. Se utilizaron las ecuaciones de Coticchia y Surace (1978) publicadas en la página web <http://www.gabrielortiz.com/index.asp?Info=058a>. El correcto funcionamiento de la conversión fue chequeado con el Global Mapper (versión 13).
2. Cálculo de la distancia entre dos puntos sucesivos (determinado trigonométricamente a partir de las coordenadas planas).
3. Medición de la longitud total de la línea, sumando la distancia entre todos los puntos sucesivos.
4. Cálculo de la equidistancia entre SP (división de la longitud total sobre la diferencia absoluta entre el SP 0 y el SP f).
5. Determinación del número de SP para cada punto intermedio. Para ello divide la longitud del intervalo sobre la equidistancia entre SP, y al resultado le suma el número del SP inicial del intervalo.
6. Generación de un archivo de salida del tipo texto ASCII, con el nombre de la línea (4° argumento), que se graba en el directorio definido en el 5° argumento.

El archivo de salida (figura 3) está conformado por un encabezado (1ª fila), que a su vez puede dividirse en dos partes, y un cuerpo (resto de las filas). La primera parte del encabezado consisten en 6 columnas: Longitud (#Long), Latitud (Lat), N° SP, Nombre de la Línea, X, Y (coordenadas planas en la proyección UTM-20). La segunda parte del encabezado contiene la equidistancia entre SP (m), la longitud total de la línea (*Length*), los datos de SP inicial (SP 0) y final (SP f), las fuentes de los datos de navegación (Fuente NAV) y de los SP (Fuente SPs), y el archivo de navegación utilizado (Archivo NAV). Esta información será luego extraída por programa BaseData_JAVA. Finalmente, el cuerpo del archivo está constituido por 6 columnas con la información descrita en la primera parte del encabezado para cada uno de los puntos que definen la línea. Las 4 primeras columnas contienen la información necesaria para cargar los datos en el Kingdom Suite.

BaseData_JAVA

El programa BaseData_JAVA es bastante sencillo y no requiere ningún argumento.

```
java -jar JAR/BaseData_JAVA.jar
```

BaseData_JAVA lee todos los archivos contenidos en el directorio (y subdirectorios) previamente generados por el programa NAV_JAVA, y extrae de cada uno el nombre de la línea y los datos de la segunda parte del encabezado, donde se describe información importante de cada línea. Los datos extraídos se guardan en un nuevo archivo (BaseData_NAVGMT.txt; Figura 4), el cual permite comparar las características de la navegación (y las respectivas fuentes consultadas) para las distintas líneas. Es esperable que una correcta determinación de la navegación se vea reflejada en, por ejemplo una equidistancia entre SP similar para las líneas sísmicas de un mismo relevamiento. De no ser el caso, se puede determinar cuál de las distintas fuentes presenta resultados (más) coherentes y rehacer el archivo de navegación.

Línea	Equidís(m)	Length(m)	SP0	SPF	FuenteNav	Fuente SPs	Archivo NAV
111-18	24,83	25329,02	1500	2520	GE	SEGY	INPUT\Tdf_GE\111-18.xyz
111	25	38080,77	40	1563	GE	SEGY	INPUT\Tdf_GE\111-k.xyz
18106	24,68	20507,85	1027	1858	GE	SEGY	INPUT\Tdf_GE\18106.xyz
18114	25,04	21182,78	1592	2438	GE	SEGY	INPUT\Tdf_GE\18114.xyz
18115	24,79	27119,03	2354	3448	GE_YPF99Map	SEGY	INPUT\Tdf_Papel\18115.xyz
18116	24,74	15064,86	3527	4136	GE_YPF99Map	SEGY	INPUT\Tdf_Papel\18116.xyz
18130	25,27	7884,42	1020	1332	GE	SEGY	INPUT\Tdf_GE\K-18130.xyz
38174-18	24,9	20744,32	975	1808	GE	SEGY	INPUT\Tdf_GE\K-38174-18.xyz
38175-18	24,76	32186,69	998	2298	GE	SEGY	INPUT\Tdf_GE\38175-18.xyz
43-18	24,67	25045,1	1436	2451	GE	SEGY	INPUT\Tdf_GE\43-18.xyz
113-18	25,05	28685,15	1430	2575	GE	SEGY	INPUT\Tdf_GE\113-18.xyz
113	25,14	36205,91	13	1453	GE	Papel	INPUT\Tdf_Papel\113.xyz
18108	26,79	31981,73	2755	3949	GE_YPF99Map	Papel	INPUT\Tdf_Papel\18108.xyz
18109	25,34	20952,44	1894	2721	GE_YPF99Map	Papel	INPUT\Tdf_Papel\18109.xyz
18110_A	26,29	13541,61	2475	2990	GE_YPF99Map	Papel	INPUT\Tdf_Papel\18110_A.xyz
18110_B	25,02	29704,48	2947	4134	GE_YPF99Map	Papel	INPUT\Tdf_Papel\18110_B.xyz

Figura 4. Fragmento del archivo generado por el programa BaseData_JAVA. Permite comparar distintos parámetros de cada línea sísmica y las fuentes consultadas. En la segunda columna se puede observar que la equidistancia entre puntos de explosión de las líneas sísmicas es similar.

Figure 4. Fragment of a file generated by BaseData_JAVA which allows to compare the different files parameters for each seismic line and the sources consulted. The second column shows a similar shot point equidistance for the different seismic lines.

FileByteFiller

Este programa fue realizado para la corrección del error tipo 1 de los Segy. Este tipo de error se caracteriza por presentar encabezado textual y binario correctos (ver anexo para una descripción de la estructura de un archivo Segy), y un desfase en un sector del perfil sísmico (Figura 5). Este efecto se atribuyó a que la traza justo antes del desfase tiene menos bytes (o datos) de los definidos en el encabezado, de manera tal que los programas de visualización sísmica (por ej. SeiSee) completan los bytes que faltan con los datos de la traza siguiente, produciendo así el desfase.

FileByteFiller requiere de 4 argumentos:

```
java -jar fileByteFiller.jar inFile: "WG-15b-A.sgy" startByte: 3600 bloques: 1660 outFile: "WG-15b-A_Fixed.sgy"
```

archivo de entrada (inFile), número de byte dañado (startByte), cantidad de bytes para completar la traza (bloques) y nombre del archivo de salida (outFile).

Para arreglar el problema, fileByteFiller lee el archivo de entrada y agrega una cadena de bytes (con valor 0) para completar la traza incompleta. La cantidad de datos faltantes en la traza dañada y su posición dentro del archivo se determinan con la asistencia de programa de visualización sísmica (por ej. SeiSee) y siguiendo la estructura de los archivos Segy (Norris y Faichney, 2002; ver anexo). El programa genera un nuevo archivo con el nombre del archivo de salida.

La correcta determinación de los parámetros se puede corroborar, además del aspecto visual de la línea, con los valores de los encabezados de las trazas, como por ejemplo el número de SP (Figura 5) o más precisamente, con los datos obligatorios de los números de traza (byte 115-116) y del intervalo de muestreo (byte 117-118; ver anexo).

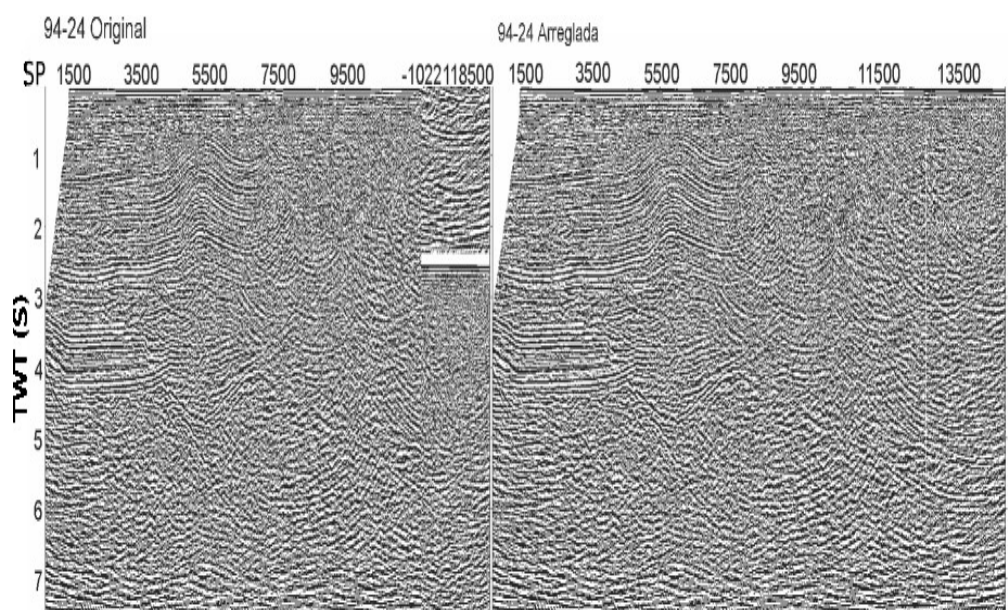


Figura 5. Imagen del programa SeiSee (versión 2.16.1) con Segy dañado correspondiente al error tipo 1. En la imagen superior se observa un desfase del perfil sísmico. Nótese que los valores de puntos de energización (SP) en el perfil inferior son coherentes con los del resto del perfil sísmico.

Figure 5. Screen capture of the SeiSee software (versión 2.16.1) of a Segy file damaged with an error type 1. The upper image shows an offset in the seismic profile. The lower image shows the seismic profile fixed with the fileByteFiller software. The shot point (SP) numbers are coherent in the lower image.

Relocate_EBCDIC_Header

El programa Relocate_EBCDIC_Header se confeccionó con el fin de arreglar el error tipo 2 en los archivos Segy, el cual se caracteriza por ser totalmente ilegibles. Es decir que en ninguna parte del Segy se observan datos esperables (Figura 6). En este caso, se asumió que los encabezados textuales y binarios estaban ubicados en una posición intermedia en el archivo Segy.

El programa funciona con 3 argumentos:

```
java -jar Relocate_EBCDIC_Header.jar inFile: "87_021" strToFind: "C 1 LINEA: " byteRewind: "0" clone
```

El primero corresponde al nombre del archivo Segy (inFile; 87_021), y el segundo a una cadena de texto (o *string*) con la cual se espera comenzaría el encabezado textual (por ejemplo "C 1 LINEA :") que está "mezclado" dentro del Segy. El programa asume que el *string* está escrito en formato EBCDIC (*Extended Binary Coded Decimal Interchange Code*), y lo convierte a formato binario para buscarlo dentro del archivo.

Una vez encontrado el *string* dentro del archivo, se determina la posición de inicio del encabezado textual y se extraen los 3600 bytes siguientes correspondientes al encabezado (ver anexo). Luego, el programa genera dos archivos Segy distintos. Para el primero utiliza los bytes localizados adelante del encabezado encontrado, y para el segundo los ubicados detrás. Para el encabezado de ambos archivos utiliza los 3600 bytes extraídos previamente. Los archivos de salida reciben el mismo nombre pero con los sufijos -A y -B, respectivamente.

Para determinar el *string* a buscar se puede: 1) consultar el encabezado textual de otros archivos Segy de la misma campaña de relevamiento sísmico, o 2) utilizar el programa gratuito SegyDetective (versión 1.0), el cual permite manualmente moverse dentro del archivo buscando cadenas de texto en formato EBCDIC o ASCII.

El tercer argumento (byteRewind) se utiliza para los casos donde el *string* no corresponde al inicio del encabezado. En esos casos, por prueba y error, se puede determinar cuántos bytes se debe retroceder para abarcar a la totalidad del encabezado.

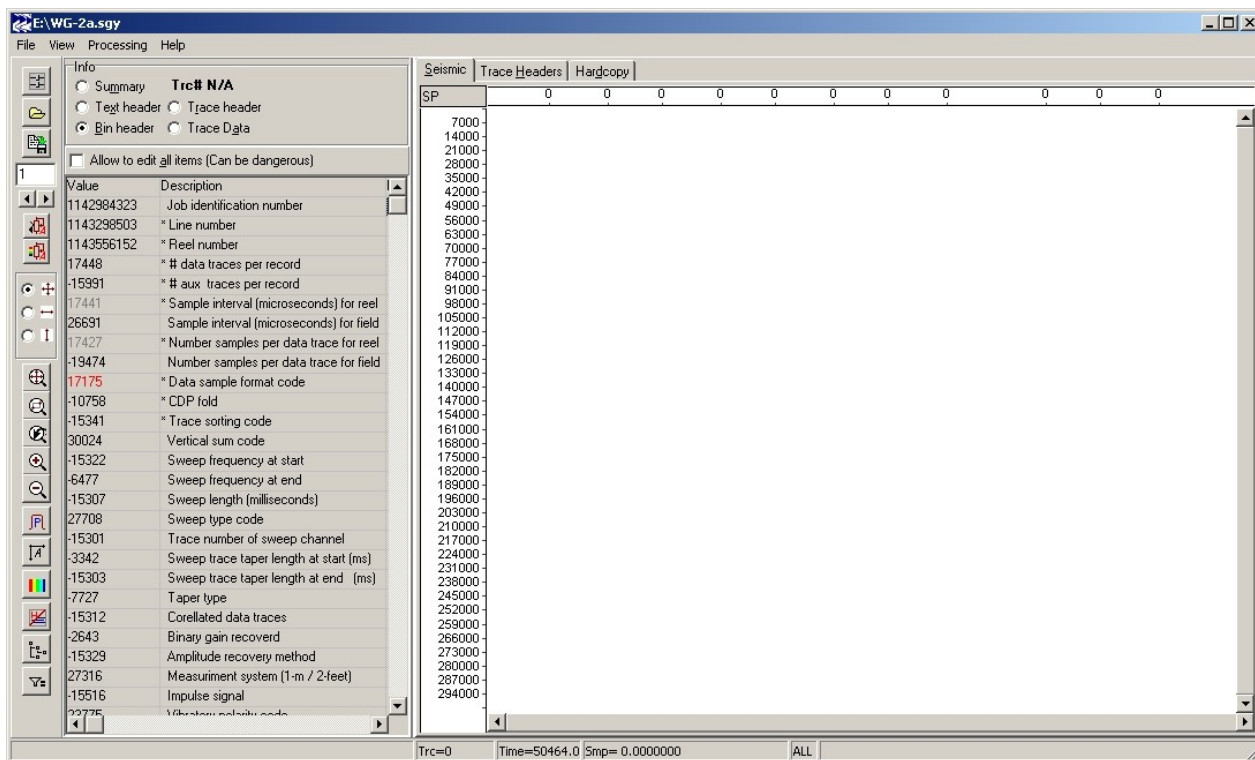


Figura 6. Imagen del programa SeiSee (V. 2.16.1) con Segy dañado con el error tipo 2. Los valores del encabezado binario y del perfil sísmico son incoherentes por lo que el programa no puede exhibir el perfil sísmico.

Figure 6. Screen capture of the SeiSee software (V. 2.16.1) of a Segy file damaged with an error type 2. The binary headers values and the seismic profiles shows incoherent values so the software failed to exhibit the seismic profile.

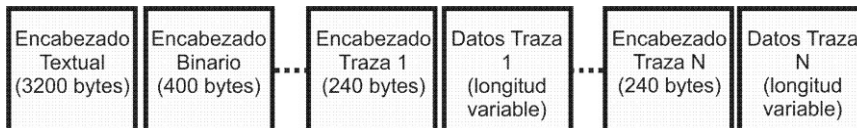


Figura 7. Esquema simplificado de la estructura de un archivo Segy. Se conforma por un encabezado textual (3200 bytes) y otro binario (400 bytes), seguido N cantidad de trazas cada una compuesta por un encabezado de la traza (*trace header*, 240 bytes) y los respectivos datos (de longitud variable). Modificado de Norris y Faichney (2002).

Figure 7. Scheme of the structure of a Segy file which is conform by a text (3200 bytes) and a binary header (400 bytes), follow by N traces each one composed by a trace header (240 bytes) and the trace data of variable length. Modified from Norris y Faichney (2002).

COMENTARIOS FINALES

Se desarrollaron 5 programas *ad hoc* escritos en JAVA para alcanzar los objetivos planteados: la integración de una base de datos de archivos de navegación, y la reparación de archivos Segy. Los programas se caracterizan por su simpleza, lo que conlleva dos ventajas fundamentales: 1) acelera notablemente el tiempo empleado para la manipulación de los datos, y 2) evita la introducción de errores aleatorios.

En particular, los programas Crear_Input_JAVA y NAV_JAVA permiten crear archivos de navegación compatibles con el GMT y con el Kingdom Suite, sin importar la rectitud o tortuosidad de los relevamientos. Por otro lado, el programa BaseData_JAVA permite comparar fácilmente los parámetros de los archivos de navegación creados, facilitando la detección de errores o incoherencias entre archivos, evaluando así las distintas fuentes consultadas para la generación de los archivos.

En cambio, con los programas fileByeFiller y EBCDIC_Relocate_Header, y mediante el estudio de la estructura de los archivos Segy, se logró corregir ambos tipos de errores identificados en los archivos dañados.

Los programas son suficientemente sencillos para que cualquiera con mínimos conocimientos de programación, pueda modificarlos y adaptarlos a sus propias necesidades. Los interesados en tener una versión de los programas pueden contactar a los autores.

Agradacimientos. Se agradece al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), por el financiamiento de la beca post doctoral del primer autor. También a los Dres. Re, Vilas y Tassone del IGeBA por sus comentarios que mejoraron el manuscrito.

BIBLIOGRAFÍA

- Coticchia, A. y Surace, L., (1978). Risoluzione di problemi geodetici con le minicalcolatrici elettroniche programmabili. *Bollettino di Geodesia e Scienze Affini*, 37 (1): 111–113.
- Esteban, F.D., (2014). Estudio geofísico-geológico del subsuelo del segmento noroccidental de la Dorsal Norte de Scotia. Argentina. Tesis Doctoral (inérita). Buenos Aires. pp 304.
- Norris, M.W. and A.K. Faichney (Eds.), (2002). SEG Y rev1 Data Exchange format. Tulsa, OK: Society of Exploration Geophysicists. http://www.seg.org/documents/10161/77915/seg_y_rev1.pdf. pp 48.
- Wessel, P., Smith W. H. F., Scharroo, R., Luis, J. y Wobbe, F., (2013). Generic Mapping Tools: Improved Version Released, *Eos Trans. AGU*, 94(45), pp 409.

ANEXO

Estructura de Archivo Segy

Los archivos Segy se dividen en tres partes principales: el encabezado textual, encabezado binario y datos de las trazas (Figura 7).

Encabezado Textual

El encabezado textual (*Text Header*), conformado por los primeros 3200 bytes del archivo, contiene 40 filas de información textual legible con el fin de proveer la descripción de la línea sísmica del archivo Segy (Norris y Faichney, 2002).

Encabezado Binario

Los 400 bytes que conforman el encabezado binario (*Bin Header*) contienen valores binarios que afectan a todo el archivo Segy, y son utilizados por los programas para la correcta interpretación de los datos. Tres de ellos son fundamentales para su correcto procesamiento: número de muestras por traza de datos, intervalo de muestreo y código del formato de los datos.

El número de muestras por traza de datos (*Number of samples per data trace*), en los bytes 21-22, indica cuantos datos conforman cada traza. El intervalo de muestreo (o *Sample interval*), ubicado en los byte 17-18, indica el intervalo medido en microsegundos (μ s) entre dos datos contiguos. La multiplicación de estos dos valores determinan la longitud total (en tiempo) del archivo Segy. Modificaciones en el número de muestras, afecta al aspecto general de la línea sísmica. Por ejemplo, si se introduce un valor de número de muestras mayor al valor original, los programas interpretan los primeros valores correspondientes a la traza siguiente como propias de la traza anterior.

Datos de las Trazas

El resto del archivo Segy contiene un número variable de registros de traza de datos, cada una precedida por un encabezado de 240 bytes.

El encabezado de la traza (*Trace Header*) contiene información que puede cambiar de traza a traza y que es necesaria para procesar, identificar y georreferenciar cada una (Norris y Faichney, 2002). Estos datos incluyen al número original registrado en el campo (byte 9-12), el punto de explosión (byte 17-20) y el número de *Common Depth Point* (CDP; 21-24). Además, dentro de cada traza se repiten, de manera obligatoria, los valores de número de muestras por traza (byte 115-116), e intervalo de muestreo en μs (byte 117-118), ya definidos en el encabezado binario.

Finalmente, luego del encabezado se localizan los datos propios de la traza, cuya cantidad de muestras está indicada en el encabezado binario y el encabezado de la traza.

Recibido: 11-06-2014

Aceptado: 07-03-2015