

**MANUAL PARA
GENERACIÓN DE RUTAS
DE ASIGNACIÓN DE
AYUDAS HUMANITARIAS
MAXIMIZANDO EL
BENEFICIO DE LAS
FAMILIAS ATENDIDAS Y
MINIMIZANDO LA
DISTANCIA DE RUTA**

INTRODUCCIÓN

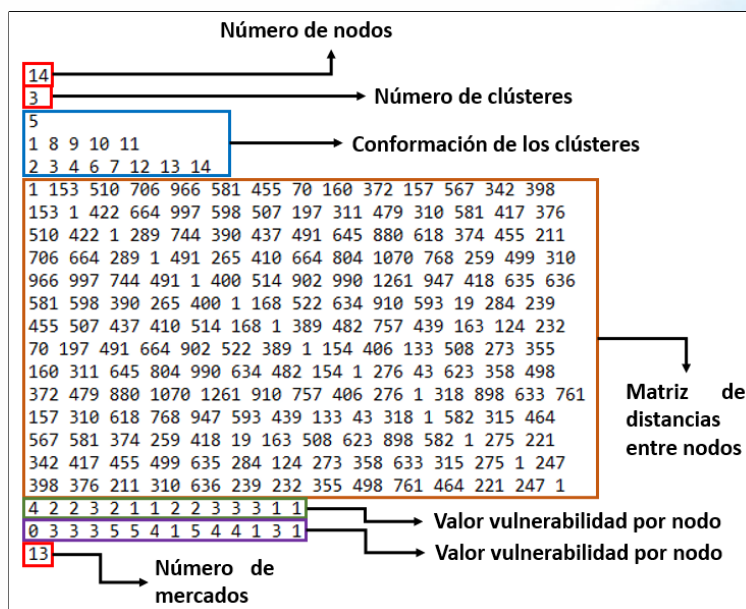
Tanto en la logística comercial como en la logística humanitaria, la generación de rutas es de gran importancia, debido a que mediante estas rutas se pretende tener los menores costos posibles o rutas más eficientes que permitan una atención de calidad en caso de emergencia. Desde el contexto del problema del agente viajero generalizado (GTSP), este manual tiene la función de ser una herramienta para la generación de rutas adecuadas, en cuanto a la atención de familias necesitadas mediante mercados durante la pandemia o en algún otro estado de emergencia que se esté presentando en cualquier lugar.

Este manual detalla cada uno de los requisitos necesarios que la organización o ente debe tener para la aplicabilidad del modelo de ruteo. También se presentan algunas recomendaciones para realizar cada uno de los pasos que este manual plantea.

REQUISITOS

Se mencionaran a continuación los requisitos necesarios antes de seguir con los pasos de implementación del modelo:

- Visual Studio.
- AMPL con licencia.
- AMPL API.
- Habilitación de librería de AMPL en C++.
- Tener los Anexos 13 y 14.
- Información del caso donde se aplicara el modelo en un archivo de texto (.txt), así:



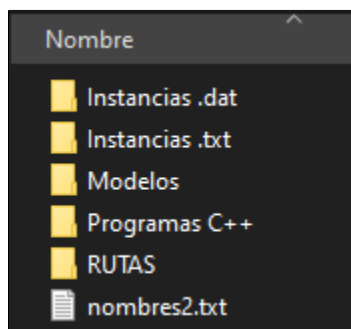
*Nota: para mayor claridad en la descarga de AMPL API, habilitación de la librería y ubicación de programas necesarios ver este video: https://www.youtube.com/watch?v=KO_H1ME1NSA.

PASOS

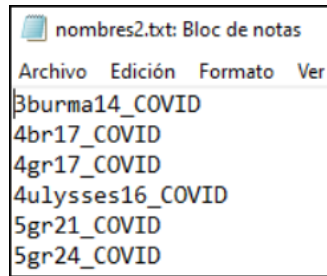
Para tener la información y archivos organizados, y cumplir con la implementación del modelo, se plantea seguir los siguientes pasos:

Paso 1: Conversión caso(s) aplicado(s) de formato .txt a .dat

En el escritorio o donde se desee, se debe crear una carpeta y dentro de ella cinco carpetas más ("Instancias .txt", "Instancias .dat", "Modelos", "Programas C++" y "RUTAS") de esta manera:

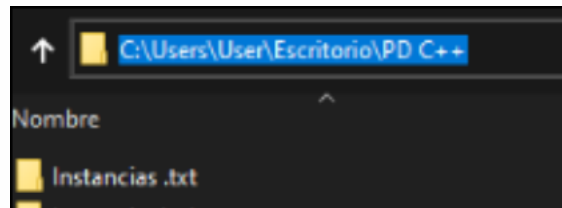


Después de haber creado estas carpetas, en la carpeta "Modelos", poner el Anexo 13 y extraer los archivos, en la carpeta "Programas C++" poner el Anexo 14 y extraer los archivos; por último, en la carpeta "Instancias .txt" poner la información del caso o casos de aplicación. Además de esto, se debe crear un archivo .txt ("nombres2") que debe contener el o los nombres de los casos de aplicación, así:



Paso 2: Implementación del programa DATOS

En la ubicación donde se crearon la carpetas, se debe copiar y extraer esa dirección, de esta manera:



Luego, en la carpeta “Programas C++” ya con los archivos extraídos del Anexo 14, se debe abrir la carpeta “DATOS” y abrir el archivo “DATOS.sln”, aparecerá la visualización del código desde la aplicación de Visual Studio. En la pestaña “Source.cpp”, en la línea 6, se debe reemplazar la ubicación que está entre comillas y poner la nueva, añadiendo un backslash, sin eliminar “\\nombres2.txt”:

```
6 fstream f("C:\\Users\\User\\Escritorio\\PD C++\\nombres2.txt", fstream::in);
```

También se debe modificar la dirección entre comillas de la línea 55, según sea la ubicación de la carpeta

“Instancias .txt”, teniendo en cuenta este nombre y los backslash:

```
55 string dire = ("C:\\Users\\User\\Escritorio\\PD C++\\Instancias .txt\\" + name + ".txt");
```

La línea 94 amerita la modificación de la dirección, según sea la ubicación de la carpeta “Instancias .dat”, teniendo en cuenta este nombre y los backslash:

```
94 ofstream k("C:\\Users\\User\\Escritorio\\PD C++\\Instancias .dat\\" + nombre + ".dat");
```

Luego pulsar el botón “Depurador local de Windows” y esperar a que el programa muestre una ventana con la palabra “fin.”. Después cerrar la ventana.

Paso 3: Implementación del programa DATOS1SQL

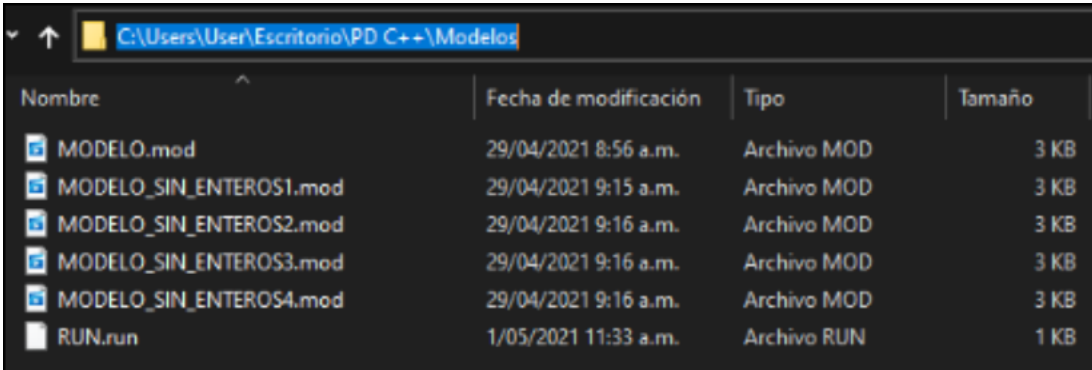
Volviendo a la carpeta “Programas C++”, abrir la carpeta “DATOS1SQL” y por consiguiente abrir el archivo “DATOS1SQL.sln”, donde se visualizará el código desde Visual Studio. En la pestaña “Source.cpp” en la línea 6, se debe poner la dirección del paso anterior:

```
6 fstream f("C:\\Users\\User\\Escritorio\\PD C++\\nombres2.txt", fstream::in);
```

En la pestaña “Header.h”; es imperativo tener cuidado con la ubicación entre comillas que está en la línea 21 (Ver video propuesto al inicio), pues esta dirección señala la

ubicación del archivo “ampl.exe”, que está junto con las carpetas de AMPL API y AMPL IDE.

En la línea 27, poner la dirección donde se encuentra el Anexo 13, que se extrajo dentro de la carpeta “Modelos”:



A screenshot of a Windows File Explorer window. The address bar shows the path 'C:\Users\User\Escritorio\PD C++\Modelos'. The main area displays a list of files and folders with columns for 'Nombre', 'Fecha de modificación', 'Tipo', and 'Tamaño'.

Nombre	Fecha de modificación	Tipo	Tamaño
MODELO.mod	29/04/2021 8:56 a.m.	Archivo MOD	3 KB
MODELO_SIN_ENTEROS1.mod	29/04/2021 9:15 a.m.	Archivo MOD	3 KB
MODELO_SIN_ENTEROS2.mod	29/04/2021 9:16 a.m.	Archivo MOD	3 KB
MODELO_SIN_ENTEROS3.mod	29/04/2021 9:16 a.m.	Archivo MOD	3 KB
MODELO_SIN_ENTEROS4.mod	29/04/2021 9:16 a.m.	Archivo MOD	3 KB
RUN.run	1/05/2021 11:33 a.m.	Archivo RUN	1 KB

La nueva dirección puesta en la línea 27, debe quedar:

```
27 | ampl.read("C:\\Users\\User\\Escritorio\\PD C++\\Modelos\\" + name2);
```

Cabe resaltar que la parte “ + name2”, debe conservarse a pesar de cambiar la dirección, además de esto, hay que asegurarse que la dirección quede con doble backslash.

También se debe modificar la dirección entre comillas de la línea 28, con la dirección de ubicación de la carpeta “Instancias .dat”, teniendo en cuenta este nombre y los backslash:

```
28 | ampl.readData("C:\\Users\\User\\Escritorio\\PD C++\\Instancias .dat\\" + name1 + ".dat");
```

La línea 41, debe tener exactamente la misma dirección que esta entre paréntesis en la línea 28.

Finalmente se pulsa el botón “Depurador local de Windows” y esperar a que el programa muestre una ventana con la palabra “fin.”. Después cerrar la ventana.

Paso 4: Implementación del programa RUTASOL

Ahora en la carpeta “RUTASOL”, se debe abrir el archivo “RUTASOL.sln” y se visualizará el código desde la aplicación Visual Studio. En la pestaña “Source.cpp”, en la línea 6 poner la misma ubicación del paso 1, de este modo:

```
6      fstream f("C:\\Users\\User\\Escritorio\\PD C++\\nombres2.txt", fstream::in);
```

En la pestaña “Header.h”, en la línea 19, se debe pegar la dirección de la carpeta RUTAS que se creó en el paso 1:

```
C:\\Users\\User\\Escritorio\\PD C++\\RUTAS
```

Es importante considerar los backslash, las comillas y la parte “+ nombre + “Ruta.txt””, la cual es obligatoria:

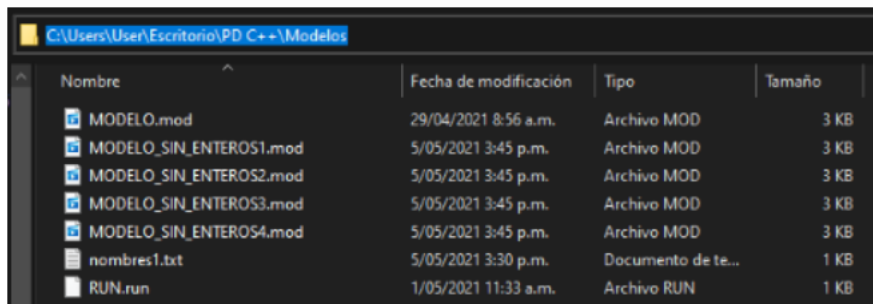
```
19      ofstream f("C:\\Users\\User\\Escritorio\\PD C++\\RUTAS\\" + nombre + "Ruta.txt");
```

En la línea 22, poner la ubicación del archivo “ampl.exe”.

Que está justo en la misma carpeta donde se encuentra el AMPL API y el AMPL IDE:

```
21 try {  
22     ampl::Environment env("D:\\Documentos\\PROGRAMAS\\AMPL\\ampl_mswin64");  
23     ampl::AMPL ampl(env);
```

En la línea 26, se debe poner la dirección de ubicación del archivo “MODELO.mod”, que se extrajo del Anexo 13:



Nombre	Fecha de modificación	Tipo	Tamaño
MODELO.mod	29/04/2021 8:56 a.m.	Archivo MOD	3 KB
MODELO_SIN_ENTEROS1.mod	5/05/2021 3:45 p.m.	Archivo MOD	3 KB
MODELO_SIN_ENTEROS2.mod	5/05/2021 3:45 p.m.	Archivo MOD	3 KB
MODELO_SIN_ENTEROS3.mod	5/05/2021 3:45 p.m.	Archivo MOD	3 KB
MODELO_SIN_ENTEROS4.mod	5/05/2021 3:45 p.m.	Archivo MOD	3 KB
nombres1.txt	5/05/2021 3:30 p.m.	Documento de te...	1 KB
RUN.run	1/05/2021 11:33 a.m.	Archivo RUN	1 KB

La dirección en la línea 26, debe quedar así:

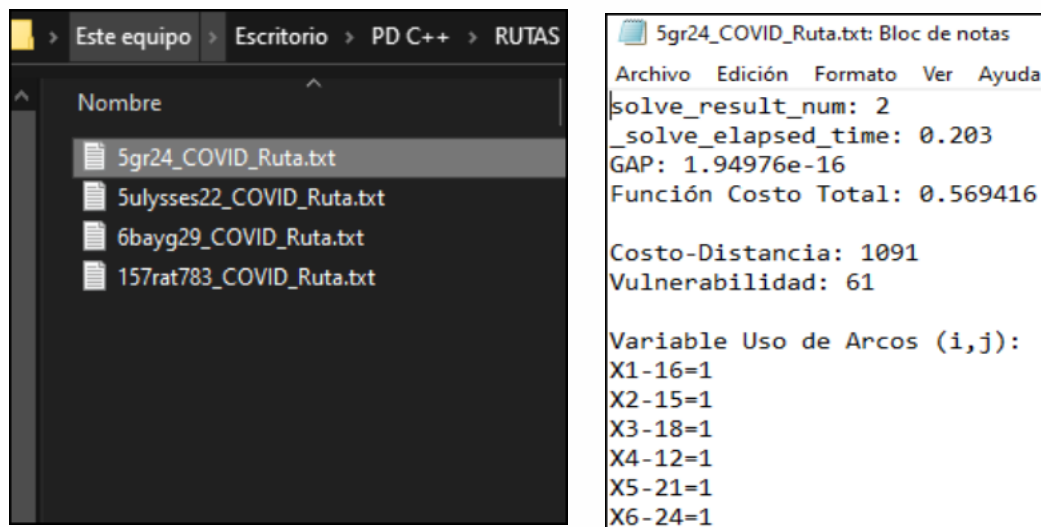
```
26 ampl.read("C:\\Users\\User\\Escritorio\\PD C++\\Modelos\\MODELO.mod");
```

En la línea 27, la nueva dirección, debe contener la ubicación de la carpeta “Instancias .dat”, similar a la siguiente, con los respectivos backslash, conservado la parte “+ name + “.dat””:

```
27 ampl.readData("C:\\Users\\User\\Escritorio\\PD C++\\Instancias .dat\\" + name + ".dat");
```

Posterior a esto, se debe pulsar el botón “Depurador local de Windows” y esperar a que el programa muestre una ventana con la palabra “fin.”. Después cerrar la ventana.

Al abrir el o los archivos que se crearon en la carpeta "RUTAS", se encuentra la información de la función objetivo multiple y las individuales, así como también el valor de las variables, las cuales muestran la ruta generada por el programa.



RECOMENDACIONES

- Se debe tener mucho cuidado con los detalles que el video propuesto explica, pues de esto depende la ejecución satisfactoria de cada uno de los programas (“DATOS”, “DATOS1SOL”, “RUTASOL”).
- Revisar detenidamente que las direcciones tengan el doble backslash, pues de esto depende que los respectivos archivos sean encontrados.
- Ceñirse a los pasos que este manual presenta.
- En el video propuesto, hay algunos ítems que en el Anexo 14 y sus carpetas ya estas contemplados, pero se recomienda también verificarlos.