



**Universidad de Santiago de Chile**  
**Facultad de Ingeniería**  
**Departamento de Ingeniería Informática**

## **Estructura de datos y análisis de algoritmos**

### **Manual de usuario**

Christofer Rodríguez

Profesor: Mario Inostroza

Ayudante: Estaban Silva

### 3.1 Introducción

El programa que usted posee actualmente no se encuentra completo, por lo que solamente le permite leer los archivos de entrada para el problema de “salvación espacial II”, para el correcto funcionamiento de este programa se deben seguir las instrucciones indicadas a continuación en este manual.

Salvación espacial II consiste en Salvación espacial II consiste en obtener el camino que permita llegar desde la tierra hasta el planeta Pizza Planet en el menor tiempo de viaje posible, esto teniendo en cuenta el combustible disponible en la nave y entregar la mejor solución, que consiste en los planetas visitados para llegar al destino, esto mediante un archivo de texto.

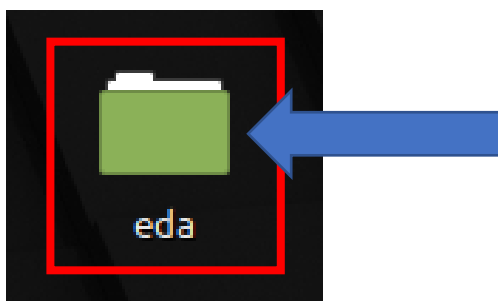
El programa entregado al usuario es el código original creada en el lenguaje de programación C, este no se encuentra completo, por lo que solamente leerá los archivos de entrada y le mostrará al usuario las estructuras creadas para almacenar estos datos.

### 3.2 Cómo compilar y ejecutar

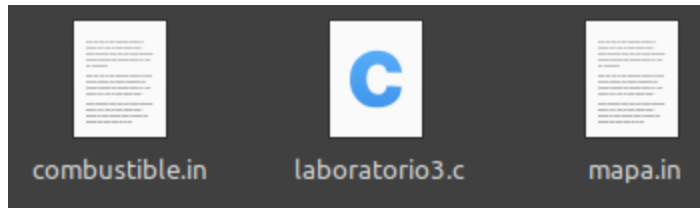
En la siguiente sección se indicará paso a paso la manera de compilar y ejecutar el programa en dos ambientes diferentes, nuevamente cabe recalcar que el programa no se encuentra completo:

#### 3.2.1 Compilación para Linux

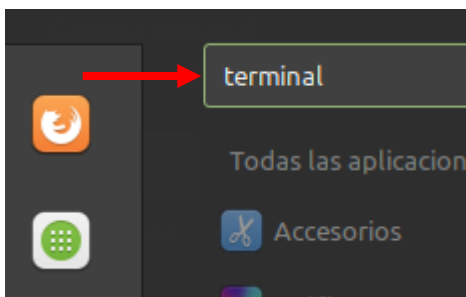
Para poder compilar el archivo, primero debe tener los archivos dentro de una carpeta.



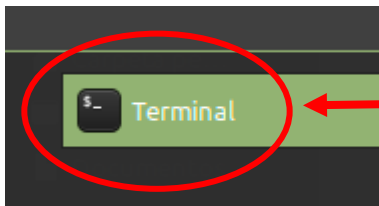
1. Como se observa en la imagen, se remarca una carpeta donde deben estar los archivos del programa, para un fácil acceso, se ha elegido el escritorio a modo de ejemplo.



2. Al abrir la carpeta debe asegurarse de que esta contenga los siguientes 3 archivos.

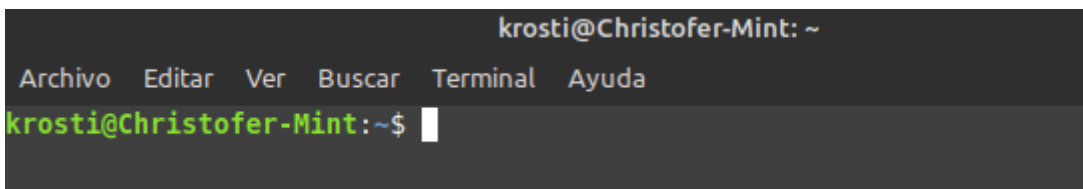


3. Una vez comprobados los archivos, necesitaremos acceder a la terminal, para esto se debe abrir el buscador y escribir la palabra "terminal".



4. Una vez aparecido el icono de la aplicación, debe clickear para abrirla.

Una vez abramos la aplicación aparecerá la siguiente ventana emergente:



5. En esta ventana emergente es la terminal, aquí debe ingresar líneas de comando para poder hacer uso de su programa.

\*Es muy importante que luego de escribir una línea de comando en la terminal, se debe presionar la tecla "enter", de esta manera se ejecutará el comando ingresado.

```
krosti@Christofer-Mint
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
krosti@Christofer-Mint:~$ cd Escritorio/eda
```

6. Ahora es necesario ingresar a la dirección donde se encuentra nuestro programa, para esto debemos escribir la palabra “cd” “espacio” seguido de la dirección donde se encuentra en este caso es “Escritorio”, “/” inmediatamente seguido del nombre de la carpeta donde se encuentran los archivos, en este ejemplo es “eda”.

Ahora que ya hemos accedido a la carpeta de los archivos desde la terminal, procederemos a compilar el programa ingresando las siguientes líneas de comando:

```
krosti@Christofer-Mint:~/Escritorio/eda$ gcc laboratorio3.c -o laboratorio3.exe
```

7. Se debe escribir “gcc” “espacio” “nombre del programa con su extensión” “espacio” “-o” “nombre del ejecutable.exe”.

Cabe destacar que el primer elemento destacado corresponde al nombre del archivo, este nombre puede variar, pero siempre debe ser igual al nombre actual del programa; mientras que el segundo elemento destacado es el nombre del ejecutable, este nombre puede ser cambiado a gusto, pero cuando se desee ejecutar, se debe hacer con el nombre dado.

Finalmente, solo resta escribir la última línea de comando para ejecutar el programa:

```
krosti@Christofer-Mint:~/Escritorio/eda$ gcc laboratorio3.c -o laboratorio3.exe
krosti@Christofer-Mint:~/Escritorio/eda$ ./laboratorio3.exe mapa.in combustible.in
```

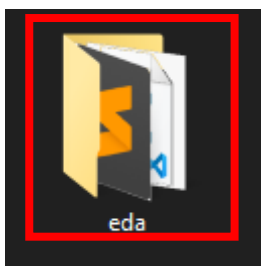
8. Aquí se debe escribir “./” inmediatamente seguido el nombre del ejecutable (en este caso es “laboratorio3.exe”) “espacio”, “nombre del primer archivo de entrada” (para este ejemplo, el archivo de entrada tiene el nombre de “mapa.in”) “espacio”, “nombre del segundo archivo de entrada” (para este ejemplo, el archivo de entrada tiene el nombre de “combustible.in”). Es totalmente necesario que los archivos se ingresen en el orden especificado en esta instrucción para el correcto funcionamiento del programa.

Ya que el programa no se encuentra completo, al ejecutarlo se mostrará mediante la consola de comandos la matriz de adyacencia creada para representar el problema y los planetas registrados. Si bien este no es el objetivo esperado del programa, esto se mostrará para ilustrar los datos leídos.

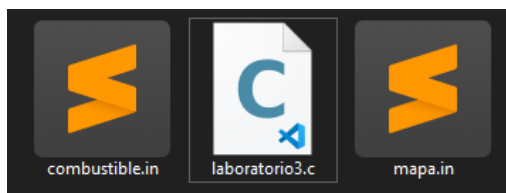
```
Nodos: 7
[0, 0] [16, 6] [10, 3] [0, 0] [0, 0] [0, 0] [0, 0]
[10, 4] [0, 0] [0, 0] [8, 4] [12, 5] [0, 0] [0, 0]
[5, 2] [0, 0] [0, 0] [15, 5] [0, 0] [15, 5] [0, 0]
[0, 0] [10, 10] [10, 5] [0, 0] [10, 5] [0, 0] [0, 0]
[0, 0] [10, 10] [0, 0] [10, 5] [0, 0] [5, 2] [10, 10]
[0, 0] [0, 0] [8, 4] [0, 0] [5, 2] [0, 0] [4, 3]
[0, 0] [0, 0] [0, 0] [0, 0] [10, 10] [4, 3] [0, 0]
El planeta: tierra tiene el indice: 0
El planeta: pandora tiene el indice: 1
El planeta: luna tiene el indice: 2
El planeta: ego tiene el indice: 3
El planeta: rigel_7 tiene el indice: 4
El planeta: axiom tiene el indice: 5
El planeta: pizza_planet tiene el indice: 6
```

### 3.2.2 Compilación para Windows

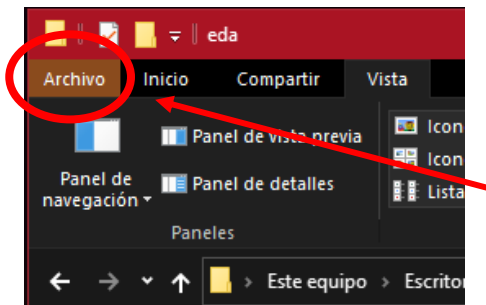
Para poder compilar en Windows, se asume que debe tener instalado MinGW y Windows PowerShell.



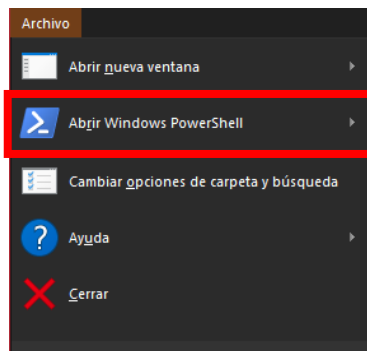
1. Para comenzar, debemos tener los archivos dentro de una carpeta, se recomienda elegir una carpeta de fácil acceso; para el caso de este ejemplo, se ha elegido una carpeta en el escritorio (Desktop).



2. A continuación, se debe abrir la carpeta y confirmar que se encuentran los siguiente 3 archivos.

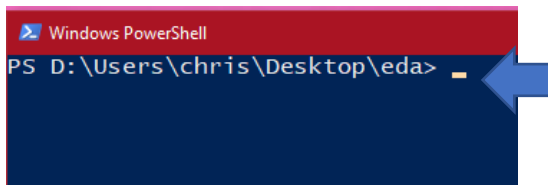


3. Una vez dentro de la carpeta, debemos presionar la pestaña llamada “Archivo”, como se puede apreciar en la imagen, esta se encuentra en la esquina superior izquierda de la ventana.



4. Una vez presionada la pestaña, se desplegará una ventana como la que se muestra en la imagen, aquí debemos presionar la opción que se muestra señalada “Abrir Windows PowerShell”.

Al seleccionar esta opción, la siguiente ventana emergente será mostrada:



5. En esta consola de comandos, debe ingresar los comandos indicados a continuación para compilar y posteriormente ejecutar el programa.

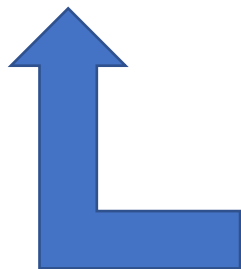
```
PS D:\Users\chris\Desktop\eda> gcc laboratorio3.c -o laboratorio3.exe
```



6. Primero se debe escribir “gcc” “espacio” “nombre del programa con su extensión” “espacio” “-o” “espacio” “nombre del ejecutable”, luego de escribir el comando, es necesario presionar la tecla “enter” para su ejecución.

Cabe mencionar que si bien, el primer nombre debe siempre corresponder al nombre del archivo (primer elemento destacado en la imagen), el segundo corresponde al nombre del archivo ejecutable (segundo elemento destacado en la imagen) y este puede ser modificado.

```
Windows PowerShell
PS D:\Users\chris\Desktop\eda> gcc laboratorio3.c -o laboratorio3.exe
PS D:\Users\chris\Desktop\eda> ./laboratorio3.exe mapa.in combustible.in
```



7. Una vez pasados unos segundos, volverá a aparecer la ruta de la carpeta (texto de la izquierda) y ahora debemos escribir “./” seguido inmediatamente del “nombre del ejecutable” (en el caso de este ejemplo es “laboratorio3.exe”) “espacio”, el nombre del primer archivo de entrada que puede tener cualquier nombre (en el caso de este ejemplo es “mapa.in”), “espacio”, “nombre del segundo archivo de entrada” (en el caso del ejemplo es “combustible.in”) y presionar la tecla “enter”.

Es totalmente necesario que los archivos se ingresen en el orden especificado en la instrucción anterior para el correcto funcionamiento del programa.

Ya que el programa no se encuentra completo, al ejecutarlo se mostrará mediante la consola de comandos la matriz de adyacencia creada para representar el problema y los planetas registrados. Si bien este no es el objetivo esperado del programa, esto se mostrará para ilustrar los datos leídos.

```
Nodos: 7
[0, 0] [16, 6] [10, 3] [0, 0] [0, 0] [0, 0] [0, 0]
[10, 4] [0, 0] [0, 0] [8, 4] [12, 5] [0, 0] [0, 0]
[5, 2] [0, 0] [0, 0] [15, 5] [0, 0] [15, 5] [0, 0]
[0, 0] [10, 10] [10, 5] [0, 0] [10, 5] [0, 0] [0, 0]
[0, 0] [10, 10] [0, 0] [10, 5] [0, 0] [5, 2] [10, 10]
[0, 0] [0, 0] [8, 4] [0, 0] [5, 2] [0, 0] [4, 3]
[0, 0] [0, 0] [0, 0] [0, 0] [10, 10] [4, 3] [0, 0]
El planeta: tierra tiene el indice: 0
El planeta: pandora tiene el indice: 1
El planeta: luna tiene el indice: 2
El planeta: ego tiene el indice: 3
El planeta: rigel_7 tiene el indice: 4
El planeta: axiom tiene el indice: 5
El planeta: pizza_planet tiene el indice: 6
```

### 3.3 Funcionalidades del programa

El presente programa es solamente capaz de leer los archivos y generar las estructuras de datos para almacenarlas, siempre y cuando se respete la siguiente estructura en los archivos de entrada:

**Primer archivo de entrada: Mapa**

```
7
tierra pandora 16 6
tierra luna 10 3
pandora tierra 10 4
```

En la primera línea del archivo debe escribirse la cantidad total de vértices (planetas) presentes en el problema. Este debe ser un número entero mayor a cero.

En las líneas siguientes del archivo debe ir primero el nombre del planeta desde donde se puede viajar y separado por un espacio el nombre del planeta al que se puede viajar desde el planeta previamente escrito en esta línea es necesario que se respete el orden origen-destino para el correcto funcionamiento, además el nombre de los planetas no puede contener espacios, se recomienda reemplazar los espacios del nombre con “-”, “\_” o el carácter de separación que estime conveniente. Separado por un espacio del planeta de destino se ingresa la cantidad de tiempo que se demora en viajar desde el planeta de origen al planeta de destino, luego separado por un espacio se encuentra el coste de combustible para realizar este desplazamiento; ambos números deben ser enteros mayores a cero y se debe respetar el orden descrito para el correcto funcionamiento.

Cada arista (posible viaje entre un planeta y otro) debe ir en una línea diferente con sus respectivos datos.

**Segundo archivo de entrada: Combustible**

```
10
ego rigel_7
```

En la primera línea del archivo debe ir la capacidad máxima del tanque de combustible de la nave, este debe ser un número entero mayor a cero.

En la segunda línea deben ir los nombres de los planetas en los que se puede recargar el combustible de la nave, estos deben ir separados por un espacio.



### 3.4 Posibles errores

Existen algunos errores que pueden ocurrir al momento de ingresar los datos en el archivo de entrada, estos se pueden evitar realizando las siguientes acciones:

En caso de que se ingrese un numero invalido dentro de la sección de cantidad planetas presentes en el problema, el programa mostrará dentro de la consola de comando un mensaje indicando que existe un error dentro del archivo de entrada y terminará su ejecución, tal como se muestra en la siguiente ilustración:

```
0
tierra pandora 16 6
tierra luna 10 3
```

Salida:

```
PS D:\Users\chris\Documents\Eda\Laborator
No hay nodos en el mapa ingresado.
```

Si esto ocurre, revise el archivo de entrada y asegúrese de que ingreso los datos respetando la estructura explicada anteriormente en este documento.

Otro error común puede existir al momento de ejecutar el programa, en caso de que el nombre del archivo de entrada sea incorrecto o no exista, se mostrara un mensaje en la consola indicando que el archivo no existe y el programa terminará su ejecución.

```
PS D:\Users\chris\Documents\Eda\Laboratorio 3>
El archivo de entrada no existe.
```

En caso de que esto ocurra, revise el nombre que tiene su archivo de entrada y vuelva a ejecutar el programa con su nombre escrito correctamente.

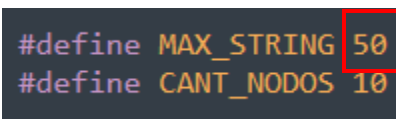
### 3.5 Limitaciones del programa

La manera en la que se implementó el programa tiene una pequeña limitación, esta puede afectar tanto a Windows como a Linux, pero la solución es la misma para ambos ambientes.

Este problema es provocado por la manera en la que se implementó la solución al problema, debido a que el lugar donde se almacena el nombre de los planetas solo es capaz de almacenar 50 caracteres.

Por lo que, si el nombre un planeta es muy largo este será muy largo como para almacenarlo, esto provocará que solamente se almacenen los primeros 50 caracteres y no el nombre completo.

Para solucionar este problema se debe abrir el archivo del programa (por defecto “laboratorio3.c”), dirigirse a la línea número 5 del documento y cambiar el valor resaltado en la siguiente imagen:

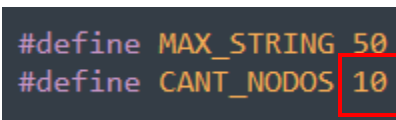


```
#define MAX_STRING 50
#define CANT_NODOS 10
```

Solamente basta con cambiar el numero resaltado por un número más alto, guardar los cambios y volver a compilar como se enseñó en este documento para poder ejecutarlo correctamente. Cabe destacar que no se debe cambiar este valor por uno muy elevado, ya que esto utilizara más recursos de su computador, se recomienda modificar este valor de ser totalmente necesario.

Debido a como se implemento este programa, existe una cantidad máxima de planetas que pueden estar presentes en el problema, en caso de que su “mapa” posea más de 10 planetas, el programa podría dejar de funcionar inesperadamente.

Para solucionar este problema es necesario abrir el archivo del programa (por defecto “laboratorio3.c”), dirigirse a la línea número 6 del documento y cambiar el valor resaltado en la siguiente imagen:



```
#define MAX_STRING 50
#define CANT_NODOS 10
```

Solamente basta cambiar el numero resaltado por la cantidad de planetas que posee su problema o por un numero mayor que le permita trabajar con todos sus problemas, luego guardar los cambios y volver a compilar como se enseñó en este documento para poder ejecutarlo correctamente. Si este valor es cambiado por uno muy elevado, puede utilizar muchos recursos de su computador, por lo que se aconseja cambiar por el número exacto de planetas.