

Estructura de Datos y Análisis de Algoritmos Manual de usuario

**Andrés Felipe Muñoz Bravo
19.646.487-5**

Profesor:
Roberto Gonzales Ibañez

Santiago - Chile
2016

TABLA DE CONTENIDOS

Tabla de Contenidos.....	3
Índice de Figuras	3
Índice de Tablas	¡Error! Marcador no definido.
CAPÍTULO 1. Introducción	4
CAPÍTULO 2. Como compilar y ejecutar	5
1.1 Compilar y ejecutar en Windows:	5
1.2 Compilar y ejecutar en Linux:.....	7
CAPÍTULO 3. Funcionalidades del programa.....	10
CAPÍTULO 4. Posibles errores	11

ÍNDICE DE FIGURAS

Ilustración 1: Abrir consola de Windows.....	5
Ilustración 2: Buscar carpeta en consola de Windows.	6
Ilustración 3: Compilar en Windows.	6
Ilustración 4: Ejecutar programa en Windows.	7
Ilustración 5: Consola en Linux.	7
Ilustración 6: Buscar carpeta en Linux.....	8
Ilustración 7: Compilar en Linux.	8
Ilustración 8: Ejecutar programa en Linux.....	9
Ilustración 9: Codificar texto.....	10
Ilustración 10: Decodificar texto.....	10

CAPÍTULO 1. INTRODUCCIÓN

Este programa es utilizado para codificar y decodificar textos. Para codificar un texto el programa lee desde un archivo de texto una línea en específico la cual contiene el texto a codificar, luego este texto pasa por una serie de algoritmos creando un nuevo texto codificado el cual es entregado en otro archivo junto con la cantidad de desfase entre las listas utilizadas para codificarlo. En el caso de decodificación, el programa realiza el proceso inverso, lee el texto codificado junto con la cantidad de desfase y el abecedario utilizado para codificarlo. De forma similar al codificado, el texto pasa por una serie de algoritmos creando un nuevo texto decodificado el cual es entregado en otro archivo de texto.

CAPÍTULO 2. COMO COMPILAR Y EJECUTAR

Tenemos que tener en claro los siguientes comandos, que son útiles dentro de la consola de Windows y de Linux.

- “cd nombre_de_carpeta” para acceder a una carpeta que exista en la dirección actual
- “cd..” para volver una carpeta atras.
- “dir” para mostrar todos los archivos y carpetas existentes en la dirección actual.

1.1 COMPILAR Y EJECUTAR EN WINDOWS:

- 1) Abrir la consola: Presionar las teclas Windows+r e ingresar cmd, luego presionar en aceptar.

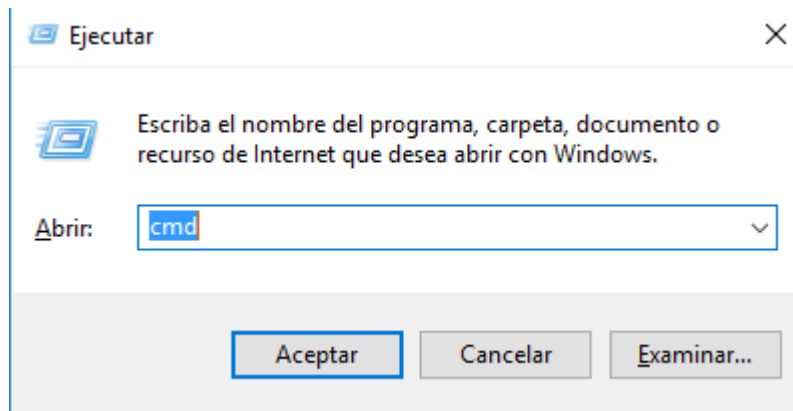
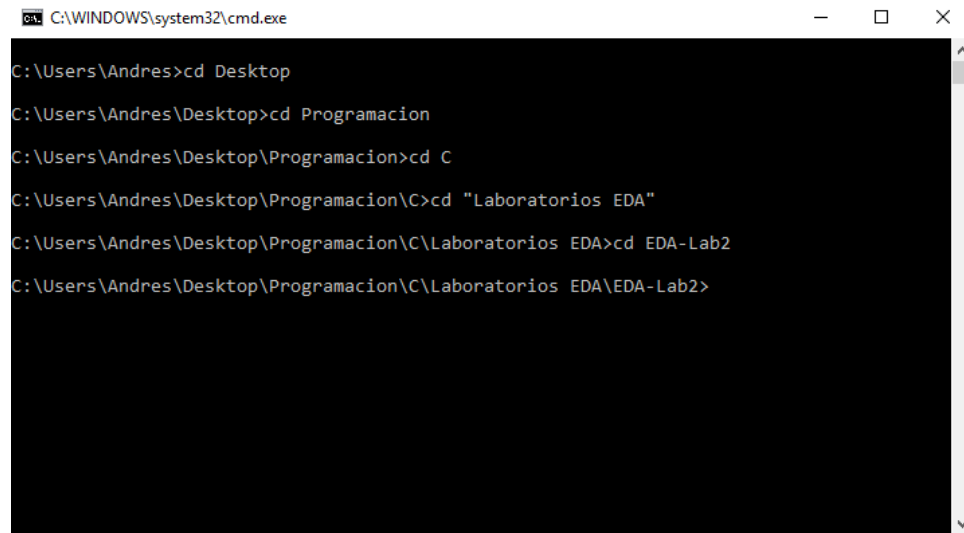


Ilustración 1: Abrir consola de Windows.

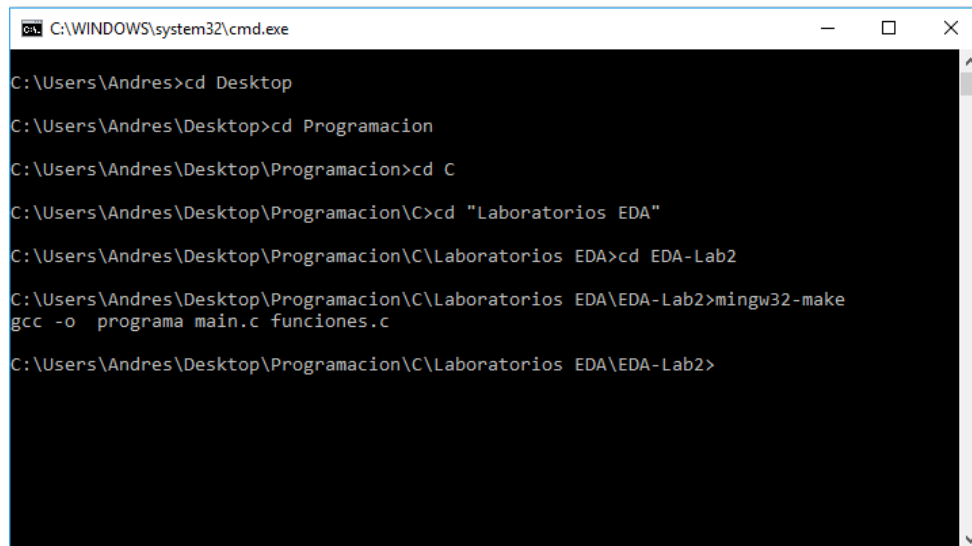
- 2) Buscar la carpeta contenedora de los archivos: Se utiliza el comando `cd` seguido de la ruta de la carpeta en la que se encuentran los archivos.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Andres>cd Desktop
C:\Users\Andres\Desktop>cd Programacion
C:\Users\Andres\Desktop\Programacion>cd C
C:\Users\Andres\Desktop\Programacion\C>cd "Laboratorios EDA"
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA>cd EDA-Lab2
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA\EDA-Lab2>
```

Ilustración 2: Buscar carpeta en consola de Windows.

- 3) Compilar los archivos: Se utiliza el comando “mingw32-make”.

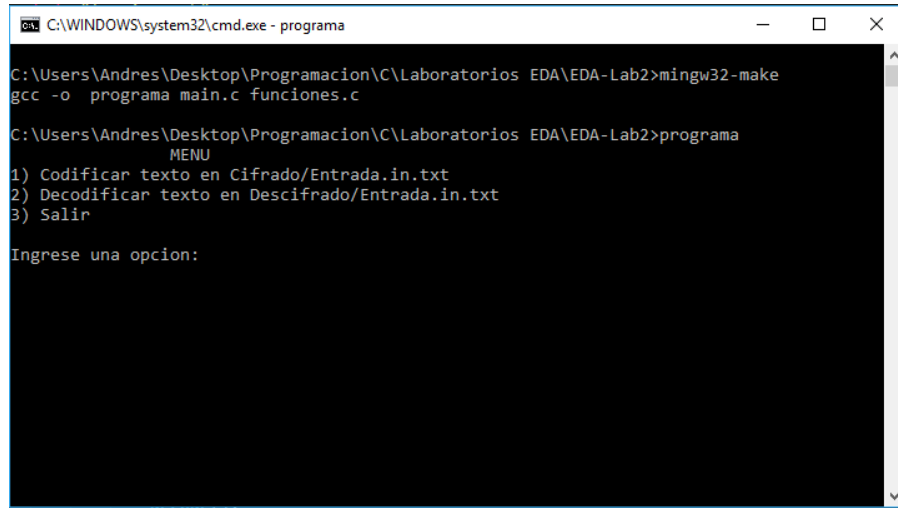


```
C:\WINDOWS\system32\cmd.exe
C:\Users\Andres>cd Desktop
C:\Users\Andres\Desktop>cd Programacion
C:\Users\Andres\Desktop\Programacion>cd C
C:\Users\Andres\Desktop\Programacion\C>cd "Laboratorios EDA"
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA>cd EDA-Lab2
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA\EDA-Lab2>mingw32-make
gcc -o programa main.c funciones.c
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA\EDA-Lab2>
```

Ilustración 3: Compilar en Windows.

Los resultados obtenidos son guardados en la carpeta “Salida”.

- 4) Ejecutar el programa: Se ingresa en la consola “programa” o doble click en el ejecutable creado al momento de compilar.



```
C:\WINDOWS\system32\cmd.exe - programa
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA\EDA-Lab2>mingw32-make
gcc -o programa main.c funciones.c
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA\EDA-Lab2>programa
MENU
1) Codificar texto en Cifrado/Entrada.in.txt
2) Decodificar texto en Descifrado/Entrada.in.txt
3) Salir
Ingrese una opcion:
```

Ilustración 4: Ejecutar programa en Windows.

1.2 COMPILAR Y EJECUTAR EN LINUX:

Se siguen prácticamente los mismos pasos que en Windows.

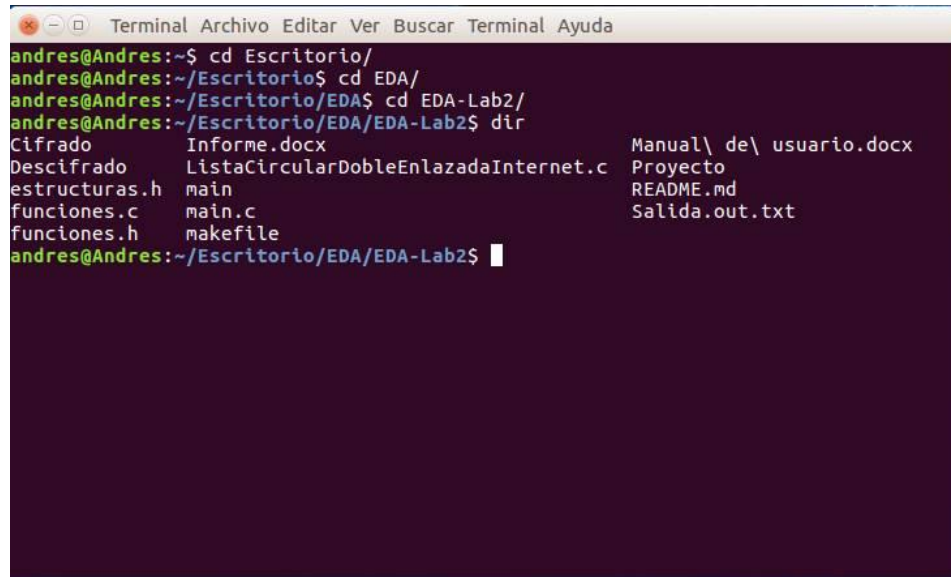
- 1) Abrir la consola: Abrir la consola o terminal desde el inicio.



```
Terminal Archivo Editar Ver Buscar Terminal Ayuda
andres@Andres:~$
```

Ilustración 5: Consola en Linux.

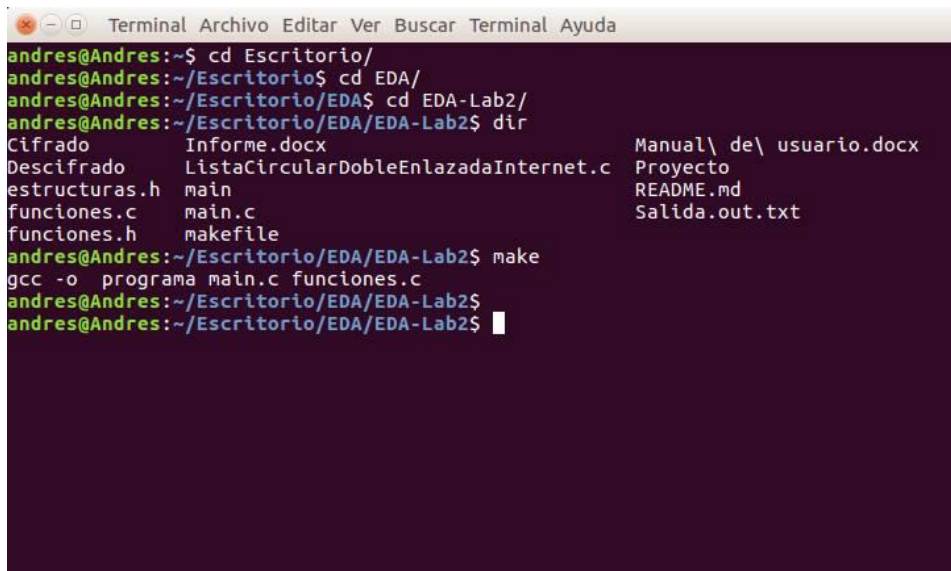
- 2) Buscar la carpeta contenedora de los archivos: Se utiliza el comando `cd` seguido de la ruta de la carpeta en la que se encuentran los archivos.



```
Terminal Archivo Editar Ver Buscar Terminal Ayuda
andres@Andres:~$ cd Escritorio/
andres@Andres:~/Escritorio$ cd EDA/
andres@Andres:~/Escritorio/EDA$ cd EDA-Lab2/
andres@Andres:~/Escritorio/EDA/EDA-Lab2$ dir
Cifrado      Informe.docx      Manual\ de\ usuario.docx
Descifrado   ListaCircularDobleEnlazadaInternet.c Proyecto
estructuras.h main              README.md
funciones.c  main.c           Salida.out.txt
funciones.h  makefile
andres@Andres:~/Escritorio/EDA/EDA-Lab2$
```

Ilustración 6: Buscar carpeta en Linux.

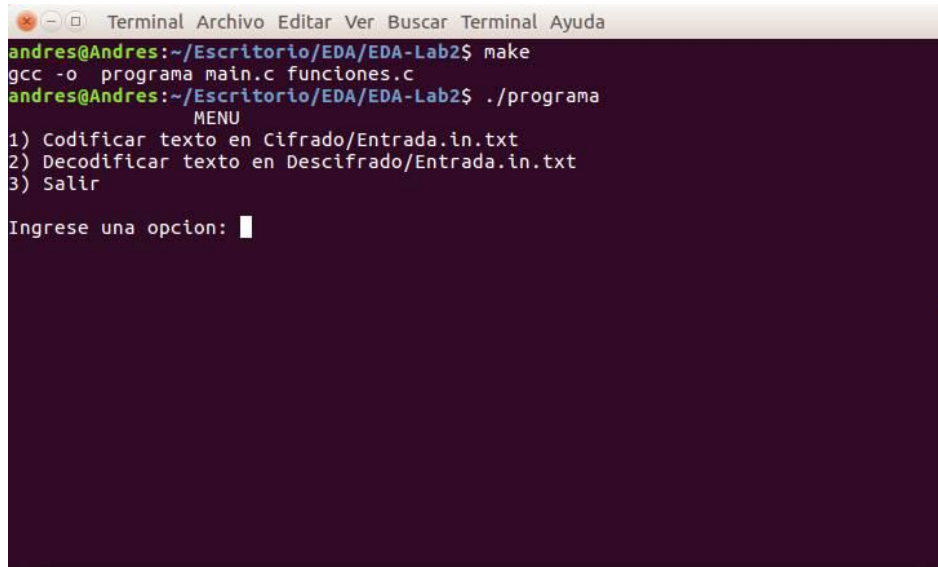
- 3) Compilar los archivos: Se utiliza el comando “`make`”.



```
Terminal Archivo Editar Ver Buscar Terminal Ayuda
andres@Andres:~$ cd Escritorio/
andres@Andres:~/Escritorio$ cd EDA/
andres@Andres:~/Escritorio/EDA$ cd EDA-Lab2/
andres@Andres:~/Escritorio/EDA/EDA-Lab2$ dir
Cifrado      Informe.docx      Manual\ de\ usuario.docx
Descifrado   ListaCircularDobleEnlazadaInternet.c Proyecto
estructuras.h main              README.md
funciones.c  main.c           Salida.out.txt
funciones.h  makefile
andres@Andres:~/Escritorio/EDA/EDA-Lab2$ make
gcc -o programa main.c funciones.c
andres@Andres:~/Escritorio/EDA/EDA-Lab2$
andres@Andres:~/Escritorio/EDA/EDA-Lab2$
```

Ilustración 7: Compilar en Linux.

- 4) Ejecutar el programa: Se ingresa en la consola “./programa”. Tener cuidado en este paso ya que no sirve solo poner “programa” como en el caso de Windows. Es necesario poner “./” antes del ejecutable.



```
Terminal Archivo Editar Ver Buscar Terminal Ayuda
andres@Andres:~/Escritorio/EDA/EDA-Lab2$ make
gcc -o programa main.c funciones.c
andres@Andres:~/Escritorio/EDA/EDA-Lab2$ ./programa
MENU
1) Codificar texto en Cifrado/Entrada.in.txt
2) Decodificar texto en Descifrado/Entrada.in.txt
3) Salir
Ingrese una opcion: █
```

Ilustración 8: Ejecutar programa en Linux.

Al igual que en Windows los resultados quedan en la carpeta “Salida”.

CAPÍTULO 3. FUNCIONALIDADES DEL PROGRAMA

Este programa tiene dos funcionalidades, una de ellas es codificar un texto que debe ser ingresado en la segunda línea del archivo de texto Entrada.in.txt, ubicado en la carpeta “Cifrado”. El texto codificado será ingresado en el archivo de texto llamado Salida.out.txt ubicado en la misma carpeta. La segunda funcionalidad es decodificar un texto del archivo Entrada.in.txt ubicado en la carpeta “Descifrado”. Para poder ingresar un texto a descifrar se debe tener en cuenta 4 cosas; La primera línea debe contener el texto “Descifrado”, la segunda línea debe contener el número de desfase entre las listas al termino de codificar el texto, la tercera línea el texto de descifrar, y en la cuarta línea las letras utilizadas para codificar el texto.

Estas funcionalidades se elegir dentro del programa como se muestra a continuación:

```

      MENU
1) Codificar texto en Cifrado/Entrada.in.txt
2) Decodificar texto en Descifrado/Entrada.in.txt
3) Salir

Ingrese una opcion: 1
El texto ha sido codificado, puedes verlo en Cifrado/Salida.out.txt
Press 'Enter' to continue: ...
```

Ilustración 9: Codificar texto.

```

      MENU
1) Codificar texto en Cifrado/Entrada.in.txt
2) Decodificar texto en Descifrado/Entrada.in.txt
3) Salir

Ingrese una opcion: 2
El texto ha sido decodificado, puedes verlo en Descifrado/Salida.out.txt
Press 'Enter' to continue: ...
```

Ilustración 10: Decodificar texto.

CAPÍTULO 4. POSIBLES ERRORES

Borrar o cambiar el nombre a los archivos de textos de entrada, estos están ubicados en la carpeta “Cifrado” y “Descifrado”, provocara que el programa deje de funcionar.

Tener un archivo de texto que dentro contenga un texto superior al tamaño de 50000 caracteres, puede provocar que el programa colapse, esto es debido a que el programa no está apto para leer líneas del tamaño formado por esta cantidad de caracteres.

Entregar un archivo con un texto que contiene símbolos o la letra ñ, será codificado pero no con el resultado esperado, esto debido a que los símbolos no pueden ser ordenados alfabéticamente.