

Estructura de Datos y Análisis de Algoritmos Manual de usuario

**Andrés Felipe Muñoz Bravo
19.646.487-5**

Profesor:
Mario Inostroza Ponta
Ayudante:
Javiera Torres Muñoz

Santiago - Chile
2016

TABLA DE CONTENIDOS

Tabla de contenido

Tabla de Contenidos.....	2
Índice de Figuras	3
CAPÍTULO 1. Introducción	4
CAPÍTULO 2. Como compilar y ejecutar	5
2.1 Compilar y ejecutar en Windows:	5
2.2 Compilar y ejecutar en Linux:.....	8
CAPÍTULO 3. Funcionalidades del programa.....	10
3.1 Cargar Imagen.	10
3.2 Mostrar características de la imagen.	11
3.3 Crear arbol según criterio de uniformidad.	12
3.4 Mostrar recorrido del arbol.	12
3.5 Mostrar regiones uniformes de la imagen.	12
3.6 Guardar archivo de los grupos uniformes	13
CAPÍTULO 4. Posibles errores.	13

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.	8
Ilustración 6: Buscar carpeta en Linux.....	8
Ilustración 7: Compilar en Linux.	9
Ilustración 8: Ejecutar programa en Linux.....	9
Ilustración 9: Funcionalidades	10
Ilustración 10: Formato archivo de texto.	10
Ilustración 11: Cargar imagen, éxito.	11
Ilustración 12: Cargar imagen, fallido.....	11
Ilustración 13: Mostrar características de la imagen.	11
Ilustración 14: Crear árbol.....	12
Ilustración 15: Recorrido de árbol.....	12
Ilustración 16: Mostrar regiones uniformes.	12
Ilustración 17: Sobrescribir un archivo.	13
Ilustración 18: No sobrescribir un archivo.	13
Ilustración 19: Guardar un archivo con nombre no existente.....	13

CAPÍTULO 1. INTRODUCCIÓN

Este programa es utilizado para analizar imágenes según un criterio de uniformidad, para esto es necesaria la implementación del tipo de dato abstracto árbol, en este caso particular se utilizan arboles cuaternarios, los cuales solo tienen 4 hijos. Estos hijos a su vez también son árboles que pueden tener más hijos y así sucesivamente. La función que cumple este tipo de dato, es poder realizar una partición de una imagen, si es que esta no cumple con la condición introducida por el usuario. De esta forma el programa crea grupos de imágenes en los cuales se cumple la condición requerida para cada uno de estos. El programa también ofrece otras funcionalidades que se presentan más adelante en este manual de usuario.

CAPÍTULO 2. COMO COMPILAR Y EJECUTAR

Hay 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 atrás.
- “dir” para mostrar todos los archivos y carpetas existentes en la dirección actual.

2.1 COMPILAR Y EJECUTAR EN WINDOWS:

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

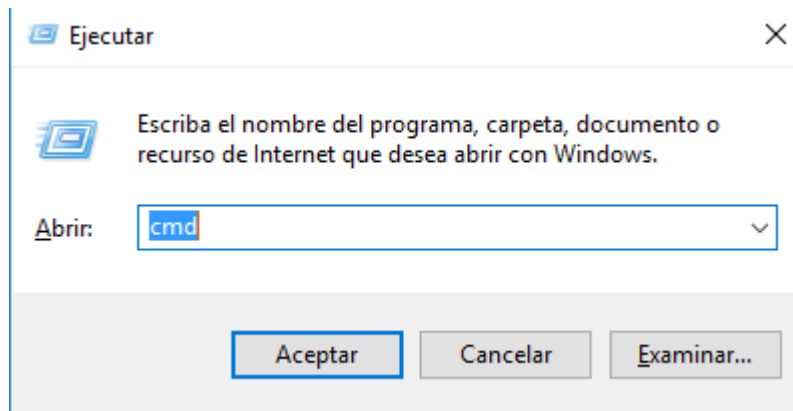
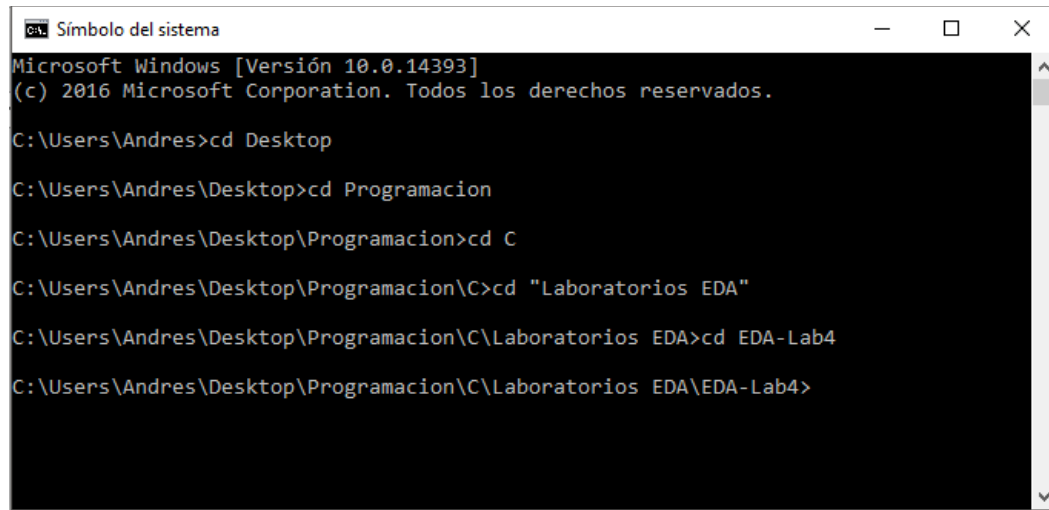


Ilustración 1: Abrir consola de Windows.

- 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.

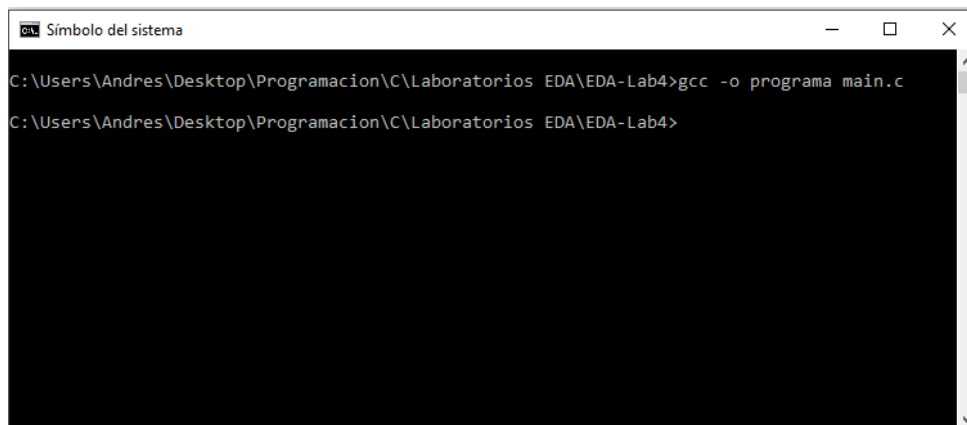


```
ca. Símbolo del sistema
Microsoft Windows [Versión 10.0.14393]
(c) 2016 Microsoft Corporation. Todos los derechos reservados.

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-Lab4
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA\EDA-Lab4>
```

Ilustración 2: Buscar carpeta en consola de Windows.

- 3) Compilar los archivos: Se utiliza el comando “`gcc -o programa main.c`”



```
ca. Símbolo del sistema

C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA\EDA-Lab4>gcc -o programa main.c
C:\Users\Andres\Desktop\Programacion\C\Laboratorios EDA\EDA-Lab4>
```

Ilustración 3: Compilar en Windows.

2.2 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.

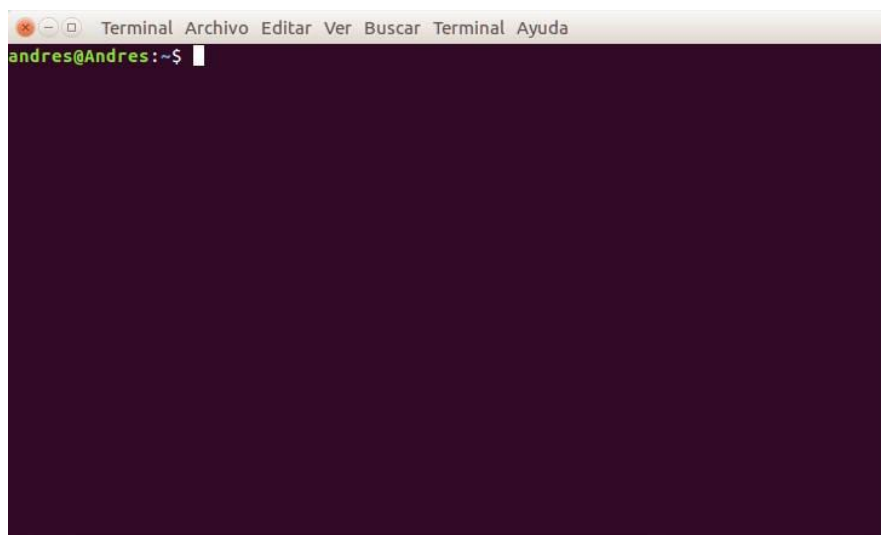


Ilustración 5: Consola en Linux.

- 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.

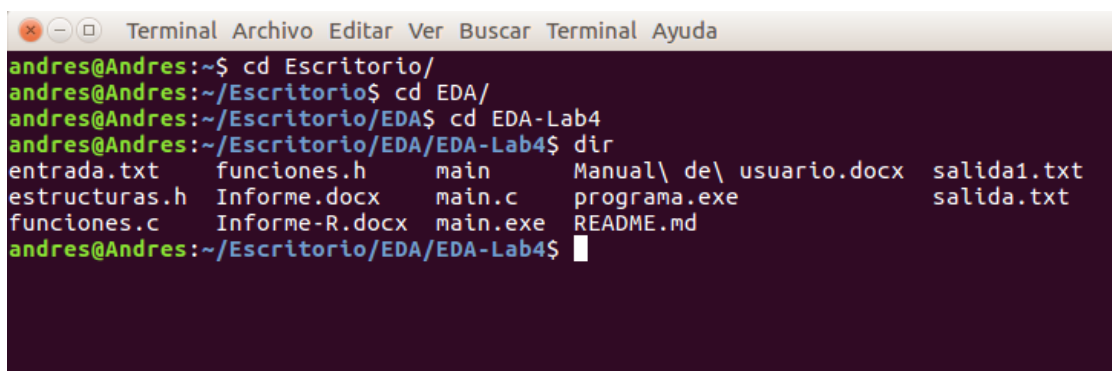


Ilustración 6: Buscar carpeta en Linux.

CAPÍTULO 3. FUNCIONALIDADES DEL PROGRAMA.

Este programa posee cinco funcionalidades que se pueden elegir dentro de este, las cuales se muestra a continuación:

```
1) Cargar imagen desde un archivo.
2) Mostrar imagen cargada.
3) Crear arbol segun criterio de uniformidad.
4) Mostrar recorrido del arbol.
5) Mostrar regiones uniformes de la imagen.
6) Guardar archivo con las regiones uniformes de la imagen.
7) Salir.
```

Ilustración 9: Funcionalidades

Para poder elegir alguna funcionalidad es necesario ingresar el número correspondiente a la funcionalidad deseada.

3.1 CARGAR IMAGEN.

Con esta funcionalidad es posible cargar una imagen desde algún archivo de texto cabe mencionar de que el archivo de texto debe contener:

1. Primera línea: dimensión de la imagen, el cual debe ser un número par, lo que significa que la imagen tiene que ser cuadrada.
2. Resto de las líneas: matriz que contiene la escala de grises (números) de cada pixel.

```
4
0 0 8 6
1 0 10 5
0 2 3 3
1 2 4 5
```

Ilustración 10: Formato archivo de texto.

Una vez elegida la opción es necesario ingresar el nombre del archivo de texto donde se encuentra la imagen con el formato anteriormente señalado.

```

Ingrese una opcion: 1
Ingrese el nombre de un archivo: entrada.txt
Imagen cargada

```

Ilustración 11: Cargar imagen, éxito.

```

Ingrese una opcion: 1
Ingrese el nombre de un archivo: entrada2.txt
El archivo ingresado no existe

```

Ilustración 12: Cargar imagen, fallido.

3.2 MOSTRAR CARACTERISTICAS DE LA IMAGEN.

Esta funcionalidad muestra por consola las siguientes características del grafo.

1. Uniformidad: Diferencia entre el mayor y el menor número dentro de la matriz.
2. Dimensión: Cantidad de filas y columnas de la imagen cargada.
3. Matriz: Matriz en la cual cada posición refiere a un pixel con su respectivo valor de gris.

Teniendo una imagen cargado previamente es posible llevar a cabo esta funcionalidad. A modo de ejemplo utilizaremos la imagen cargada en la funcionalidad anterior.

```

Uniformidad: 10
Dimension: 4
Matriz:
0 0 8 6
1 0 10 5
0 2 3 3
1 2 4 5

```

Ilustración 13: Mostrar características de la imagen.

3.3 CREAR ARBOL SEGÚN CRITERIO DE UNIFORMIDAD.

Esta funcionalidad crea un árbol según el criterio de uniformidad ingresado por el usuario.

```
Ingrese criterio de uniformidad: 2
Arbol creado utilizando criterio de uniformidad 2.
```

Ilustración 14: Crear árbol.

3.4 MOSTRAR RECORRIDO DEL ARBOL.

Esta funcionalidad muestra por pantalla como se recorre un árbol dentro de la computadora.

<pre>Ingrese una opcion: 4 Uniformidad: 10 Dimension: 4 Matriz: 0 0 8 6 1 0 10 5 0 2 3 3 1 2 4 5 Uniformidad: 1 Dimension: 2 Matriz: 0 0 1 0</pre>	<pre>Uniformidad: 5 Dimension: 2 Matriz: 8 6 10 5 Uniformidad: 0 Dimension: 1 Matriz: 8 Uniformidad: 0 Dimension: 1 Matriz: 6</pre>	<pre>Uniformidad: 0 Dimension: 1 Matriz: 10 Uniformidad: 0 Dimension: 1 Matriz: 5 Uniformidad: 2 Dimension: 2 Matriz: 0 2 1 2 Uniformidad: 2 Dimension: 2 Matriz: 3 3 4 5</pre>
--	---	---

Ilustración 15: Recorrido de árbol

3.5 MOSTRAR REGIONES UNIFORMES DE LA IMAGEN.

Esta funcionalidad muestra por pantalla una matriz de la misma dimensión de la imagen principal, lo particular de esta es que cada número representa un grupo al cual pertenece. Utilizando como ejemplo la imagen cargada para este manual se muestra a continuación esta funcionalidad.

```
Ingrese una opcion: 5
Criterio de uniformidad aplicado: 2
21 21 18 18
21 21 16 17
21 21 20 20
21 21 20 20
```

Ilustración 16: Mostrar regiones uniformes.

3.6 GUARDAR ARCHIVO DE LOS GRUPOS UNIFORMES

Esta funcionalidad ofrece al usuario guardar en un archivo de texto el resultado del programa al aplicar un criterio de uniformidad a una imagen, además pregunta al usuario si desea sobrescribir un archivo que ya existe.

```

Ingrese una opcion: 6

Ingrese un nombre para el archivo sin extencion: salida
El archivo ya existe, desea sobrescribir? (s/n): s
El archivo ha sido guardado.

```

Ilustración 17: Sobrescribir un archivo.

```

Ingrese una opcion: 6

Ingrese un nombre para el archivo sin extencion: salida
El archivo ya existe, desea sobrescribir? (s/n): n
El archivo NO ha sido guardado.

```

Ilustración 18: No sobrescribir un archivo.

```

Ingrese una opcion: 6

Ingrese un nombre para el archivo sin extencion: salida1
El archivo ha sido guardado.

```

Ilustración 19: Guardar un archivo con nombre no existente.

CAPÍTULO 4. POSIBLES ERRORES.

El único error que se puede encontrar en el programa es que no funciona correctamente cuando se ingresa un grafo sin el formato predefinido anteriormente. Como no se encontraron más errores dentro del programa se enumerarán los errores solucionados:

1. Que existan espacios antes del final del salto de línea en el archivo de texto entregado.
2. Al ingresar un nombre de archivo inválido el programa sigue funcionando.
3. El programa detecta cuando no están ingresados los datos necesarios para seguir funcionando correctamente.