



Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Estructuras de datos - IC – 2001

I semestre del 2022

Integrantes:

Kendall Guzmán Ramírez

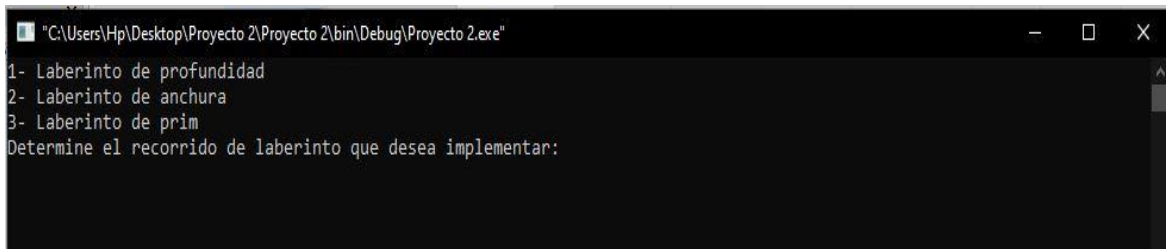
Ian Murillo Campos

Andy Porras Romero

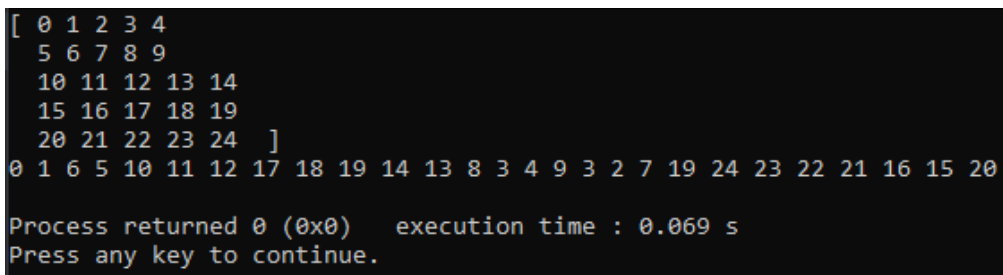
Manual de usuario:

El siguiente proyecto tiene como función crear un laberinto a partir de 3 algoritmos de búsqueda distintos (en profundidad, en anchura, Prim/Kruskal), esto lo consigue mediante el ordenamiento de los punteros de los nodos de la matriz conforme a cada algoritmo de búsqueda, asimismo también se da la opción de conocer la ruta más corta de un punto A al final del laberinto mediante otro algoritmo de búsqueda (Dijkstra/Floyd), el laberinto se representa por medio de una interfaz gráfica, al igual que el punto donde se encuentre el usuario y el final del laberinto generado, al finalizar el laberinto se generará inmediatamente después, mediante un algoritmo de búsqueda aleatorio otro laberinto de mayor tamaño y consecuentemente mayor dificultad, esto hasta que se acabe el tiempo el cual empieza a correr desde que se inicia el programa y se penaliza con una disminución del tiempo el utilizar la ayuda de la ruta más corta.

Nuestra versión del proyecto funciona de la siguiente manera, se establecen la cantidad de filas y de columnas que se desean en la matriz, luego de esto a dicha estructura se le aplican los diferentes algoritmos de búsqueda a petición del usuario y se imprimen sus conexiones en la consola.



```
"C:\Users\Hp\Desktop\Proyecto 2\Proyecto 2\bin\Debug\Proyecto 2.exe"
1- Laberinto de profundidad
2- Laberinto de anchura
3- Laberinto de prim
Determine el recorrido de laberinto que desea implementar:
```



```
[ 0 1 2 3 4
 5 6 7 8 9
10 11 12 13 14
15 16 17 18 19
20 21 22 23 24 ]
0 1 6 5 10 11 12 17 18 19 14 13 8 3 4 9 3 2 7 19 24 23 22 21 16 15 20

Process returned 0 (0x0)   execution time : 0.069 s
Press any key to continue.
```

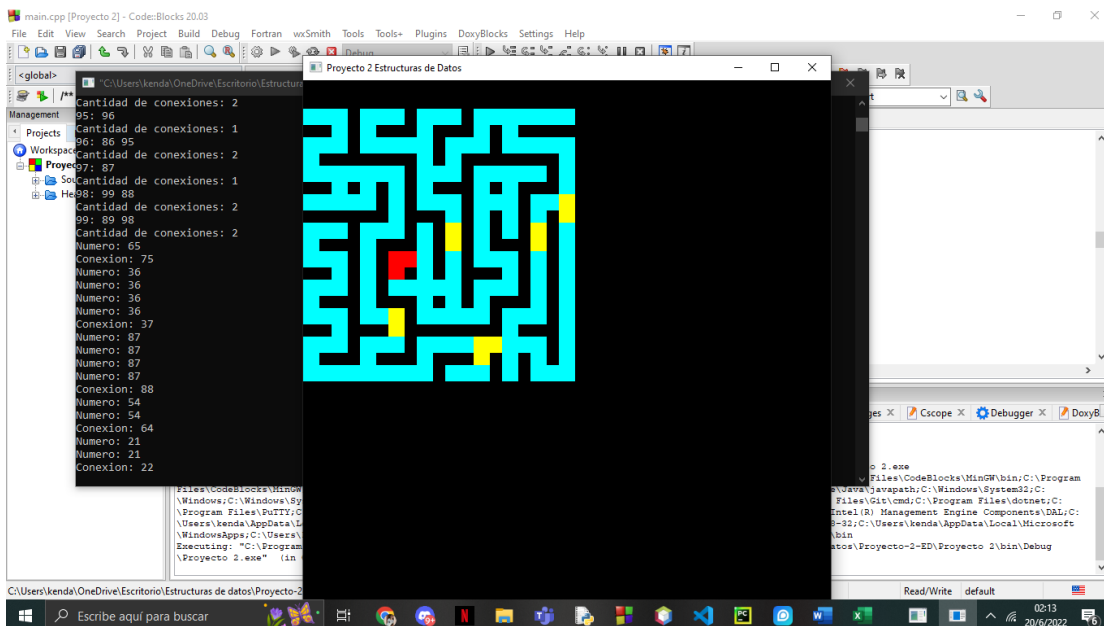
```
[ 0 1 2 3 4
  5 6 7 8 9
 10 11 12 13 14
 15 16 17 18 19
 20 21 22 23 24 ]
0 5 10 15 16 11 6 7 12 13 8 9 14 19 18 23 22 21 20 22 17 23 24 9 4 3 2 1
2 1 3 7 0 6 8 4 12 5 11 13 9 17 10 16 14 18 22 15 21 19 23 20 24

Process returned 0 (0x0)   execution time : 0.084 s
Press any key to continue.
```

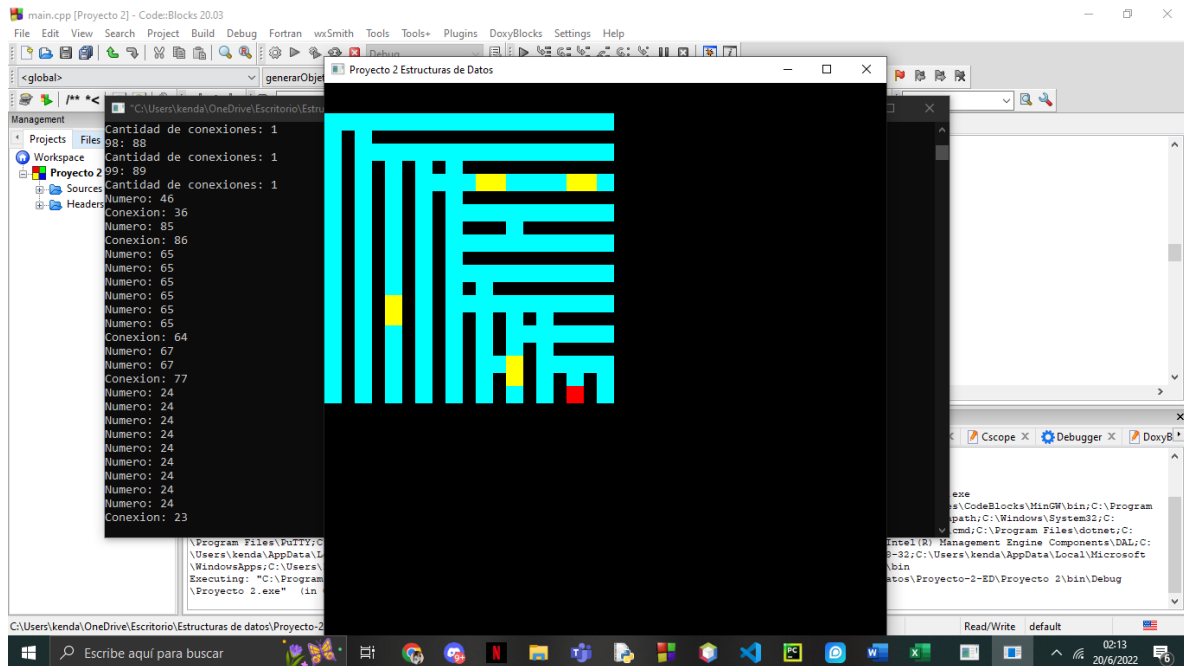
Luego de eso se crea la interfaz de usuario y esta muestra los laberintos producto de las conexiones de nodos en los grafos mediante el respectivo algoritmo de búsqueda seleccionado junto con una coloración amarilla en algunos nodos que indica donde se encuentran diferentes objetos que puede tomar el jugador y una coloración roja en donde se ubica el final del laberinto.

Ejemplos:

1. Algoritmo de búsqueda por profundidad:



2. Algoritmo de búsqueda en anchura:



3. Algoritmo de búsqueda Prim:

