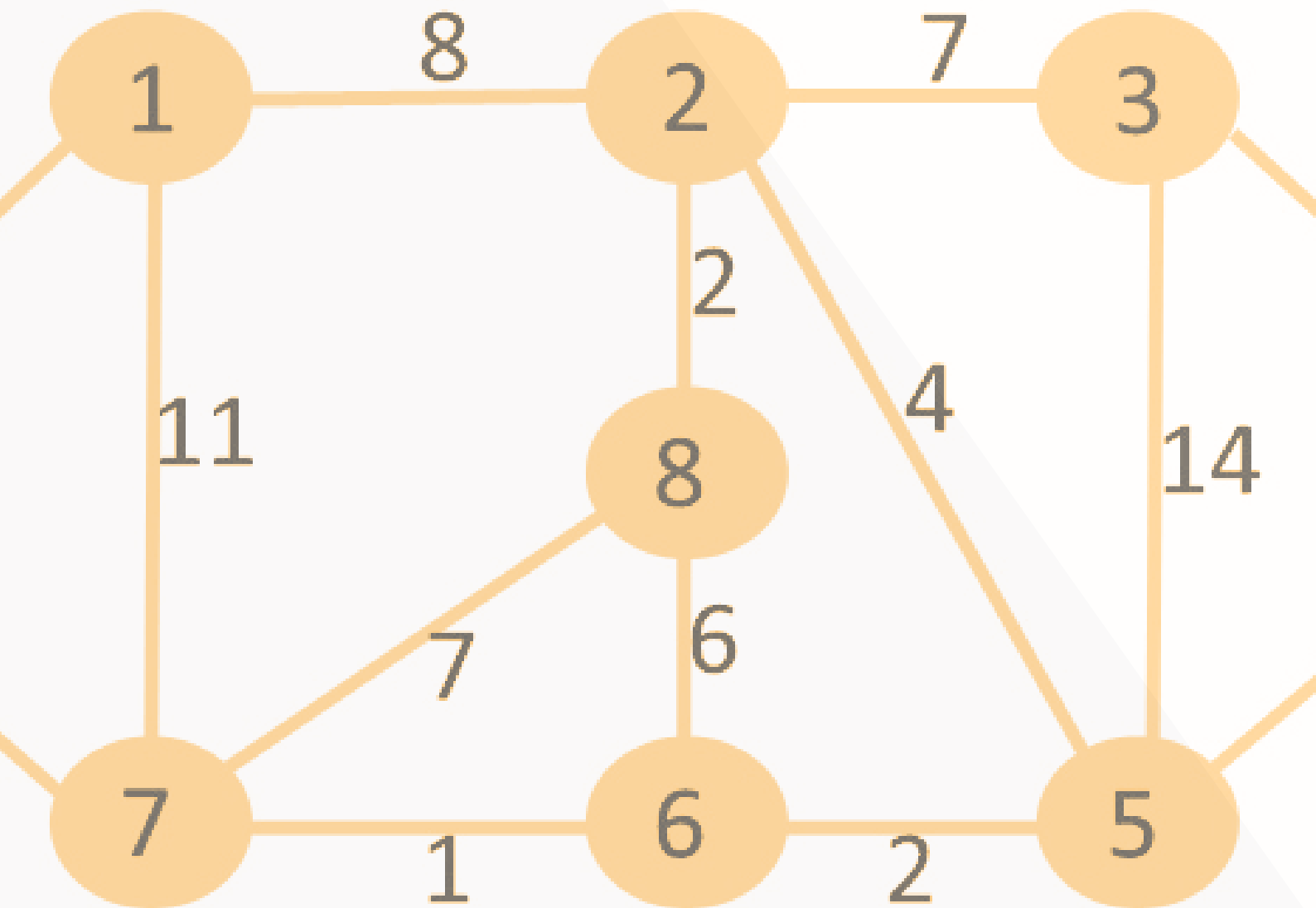


MANUAL DE USUARIO DE ALGORITMO DE DIJKSTRA

GRUPO NO. 5



ALGORITMO DE DIJKSTRA

¿QUÉ ES?

Es un modelo que se clasifica dentro de los algoritmos de búsqueda, el propósito de este algoritmo es definir cuál es la ruta más corta. Su metodología se basa en repeticiones, esto hace que en la práctica, su desarrollo se dificulte a medida que el tamaño de la red va aumentando.

La idea principal del algoritmo de dijkstra es ir explorando todos los caminos más cortos, partiendo desde el vértice de origen hacia los demás vértices, el algoritmo se detiene cuando se encuentra el camino más corto desde el vértice origen hacia el resto de los vértices del grafo.

VENTAJAS DE C++

- Fácil de instalar.
- Podemos trabajar en la versión Online de C++
- Es de código abierto
- Nos indica si hay algún error y cual es para poder corregirlo.

REQUISITOS PARA INSTALAR DEV C++

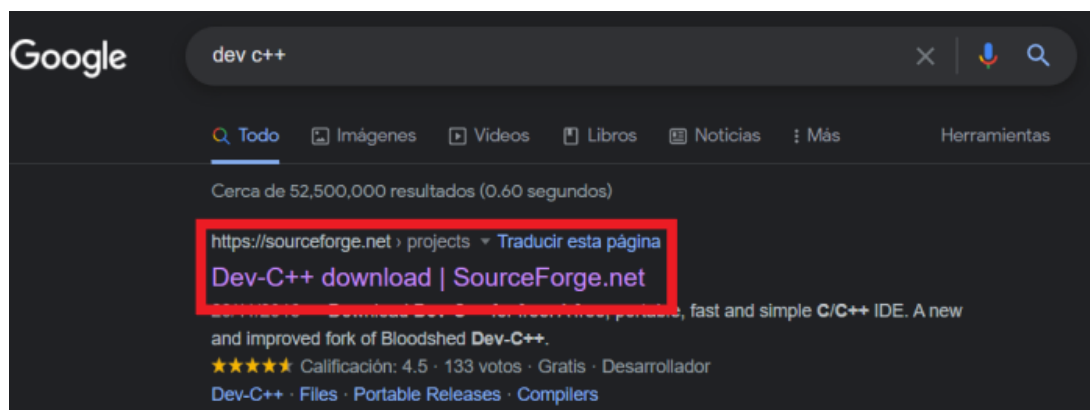
- **MEMORIA RAM:** Mínimo 1 GB.
- **SISTEMA OPERATIVO:** Sistema Operativo Windows 10 Enterprise (2016 LTSC 1607, LTSC 2019), Windows 10 (IoT Enterprise*, Home Edition**, Pro), Windows 8.1 (32 bits, 64 bits y Embedded Edition), Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2 (Standard y Datacenter Edition)
- **ESPACIO LIBRE EN DISCO:** Instalación nueva 572 MB, Actualizar, 350 MB

COMO INSTALAR DEV C++

1. Abriremos nuestro navegador de preferencia (Google Chrome, Safari, Mozilla Firefox, Microsoft Edge, etc.) y colocaremos en la barra de búsqueda "DEV C++" y presionaremos la tecla Enter.



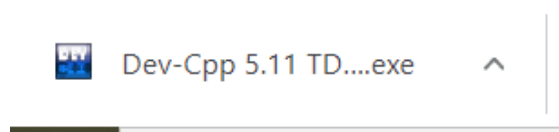
2. Daremos click en la primera opción que nos aparece.



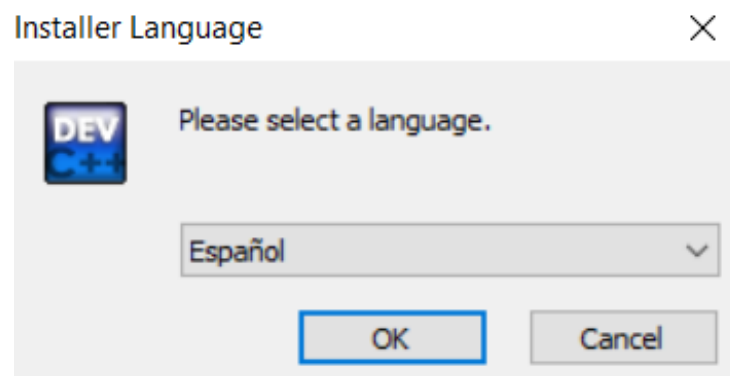
3. Nos aparecerá la siguiente pantalla, daremos click en el botón verde que dice "Download" o "Descargar".



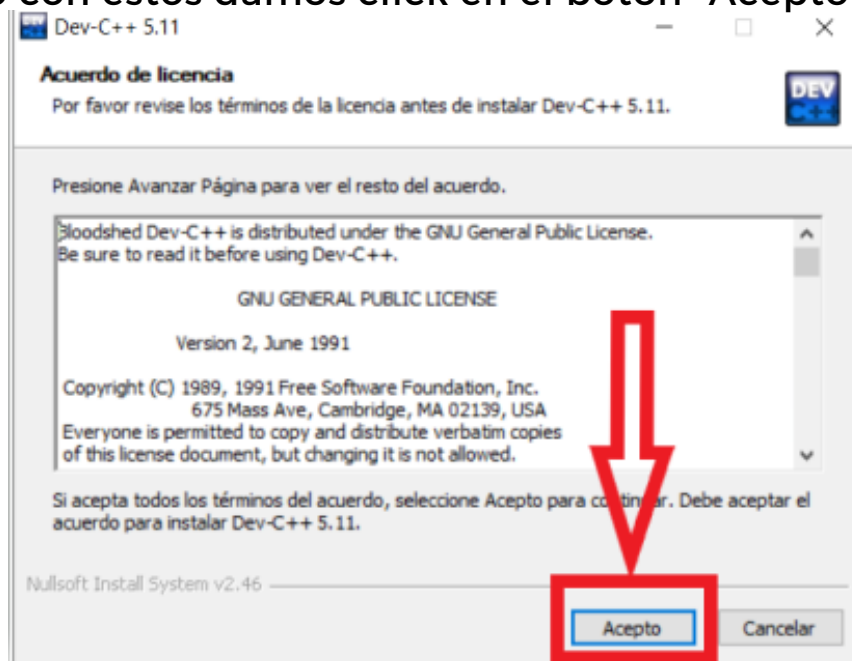
4. Esperamos algunos segundos y al finalizar la descarga nos aparecerá este icono, damos click sobre este.



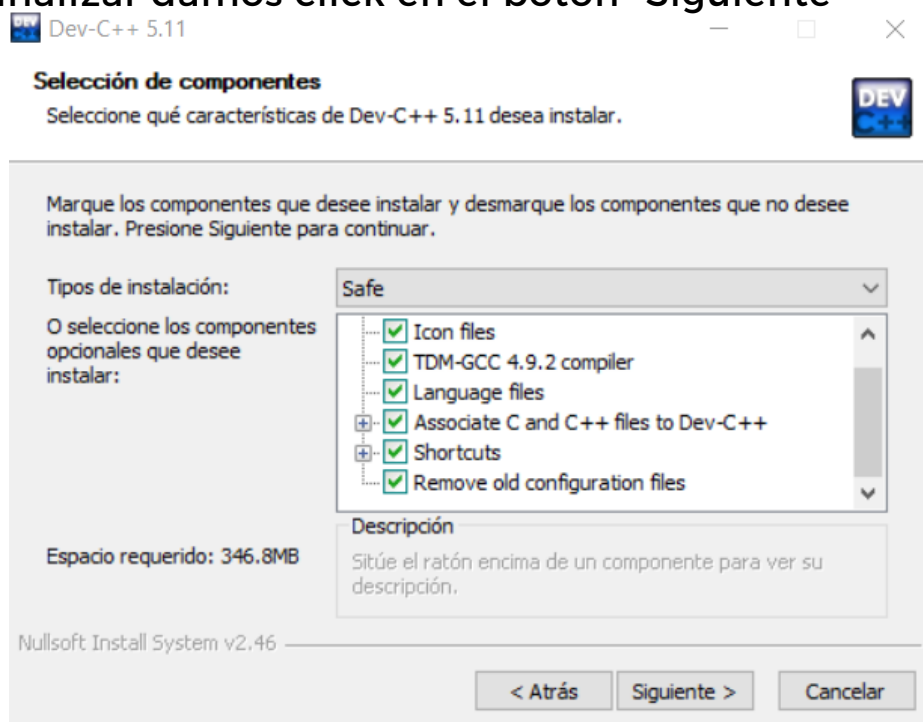
5. Nos aparecerá este icono, debemos escoger el lenguaje con el que trabajaremos.



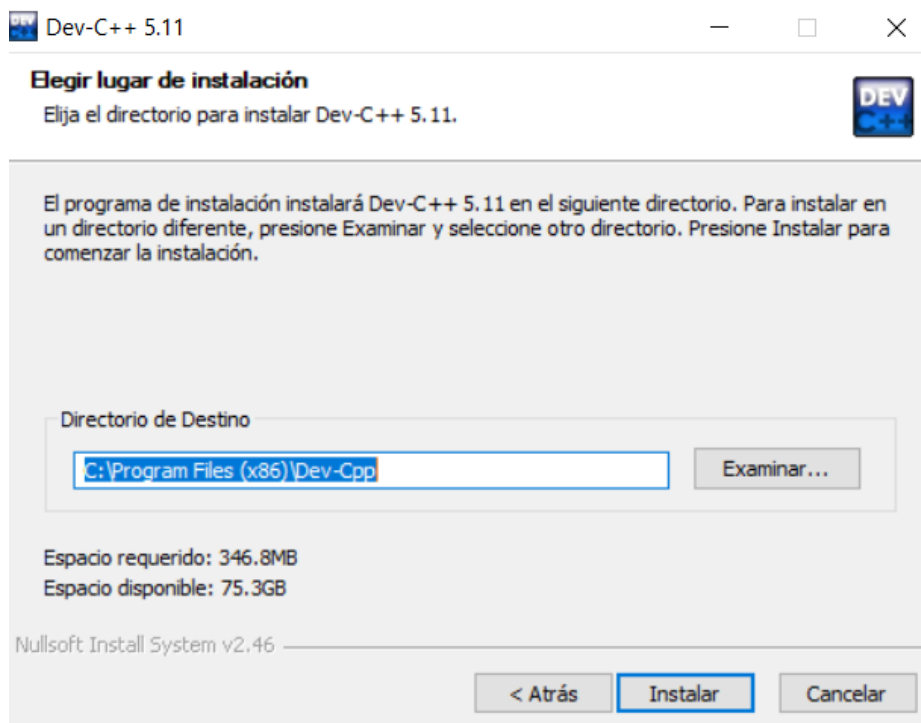
6. Leemos los términos de aceptación y si estamos de acuerdo con estos damos click en el botón "Acepto".



7. Seleccionamos los componentes que utilizaremos. y al finalizar damos click en el botón "Siguiente"



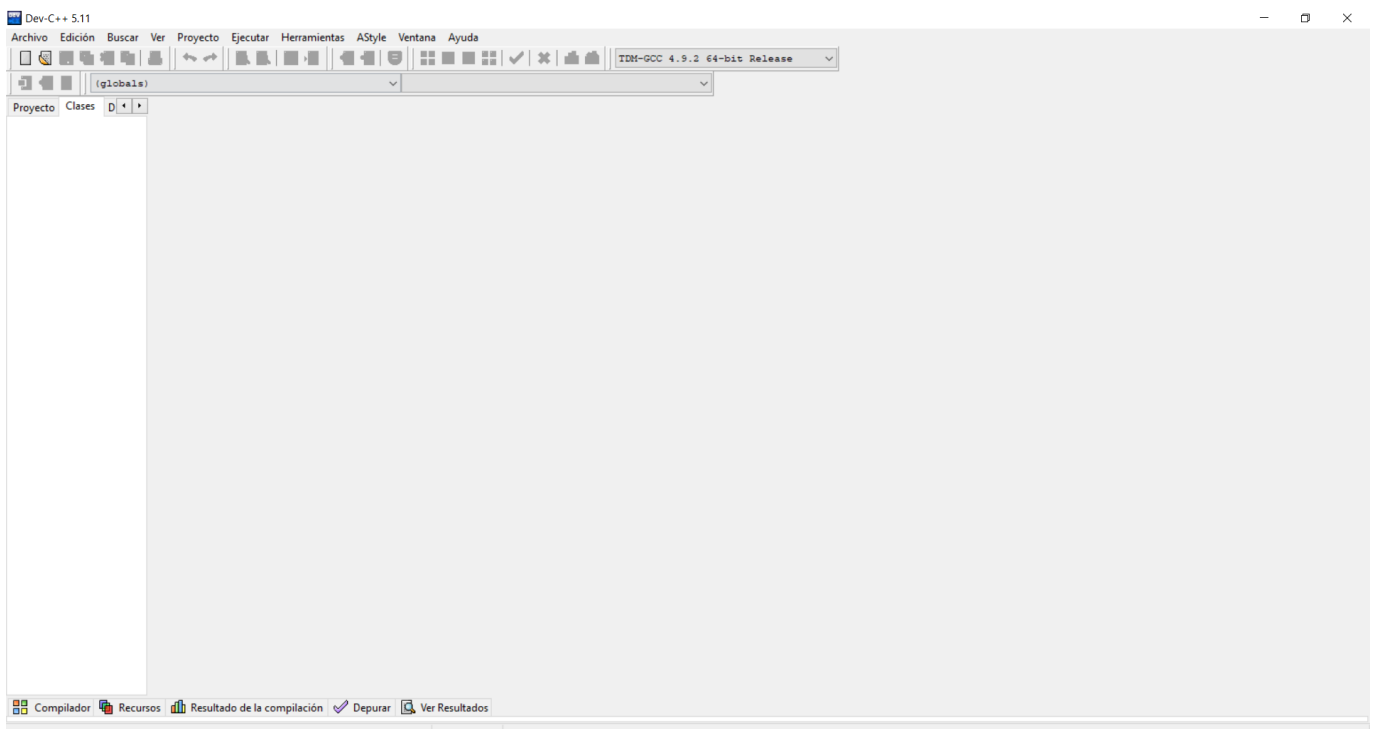
8. Colocamos la dirección en la que se encontrará ubicada la aplicación y damos click en el botón "Instalar".



9. Por último, nos aparecerá esta pantalla, damos click en el botón "Terminar" y nuestro programa iniciara su instalación.

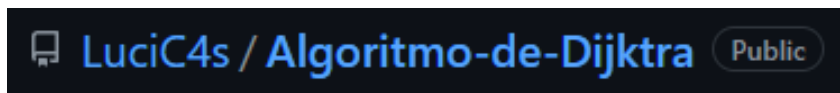


10. Al finalizar la instalación, el programa se iniciará de manera automática y nos mostrará la siguiente pantalla:

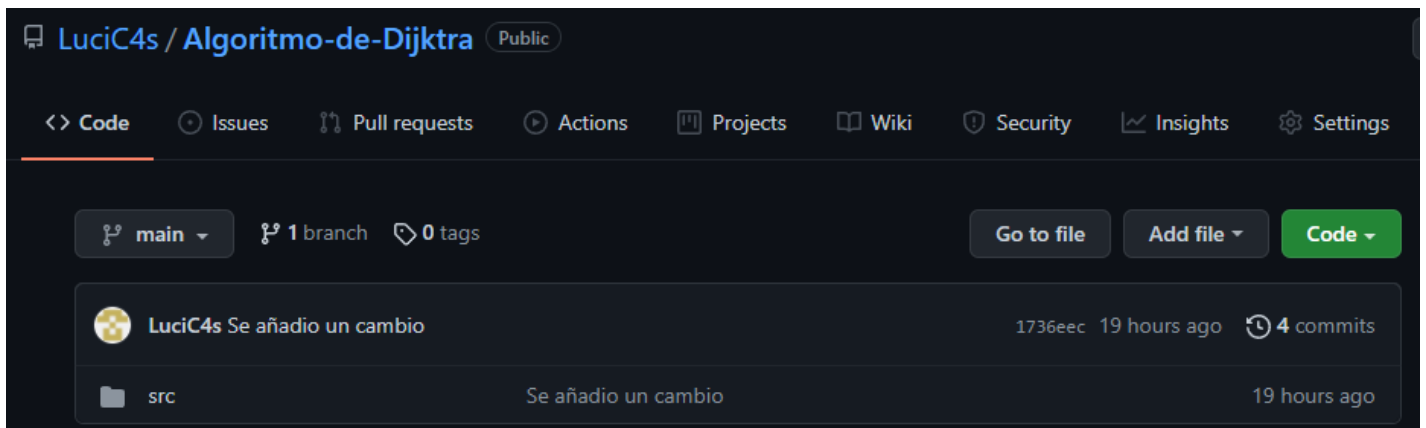


IMPLEMENTACIÓN DIJKSTRA EN C++

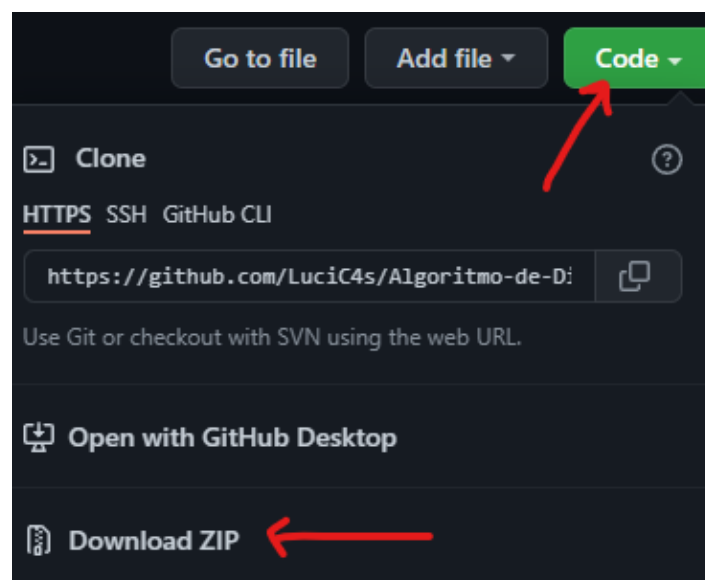
1. Dirigirse a GitHub y buscar el siguiente repositorio:



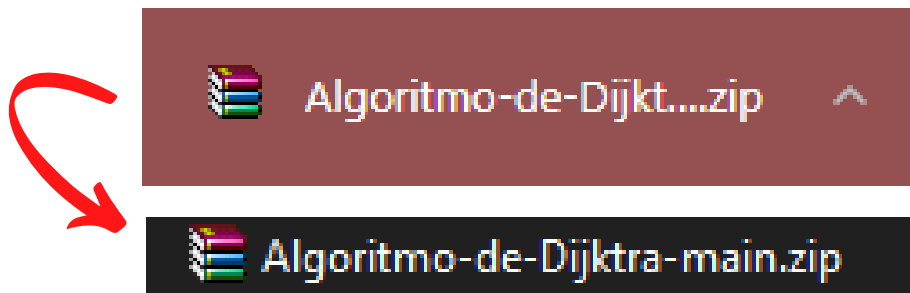
2. Aparecerá lo siguiente en pantalla:



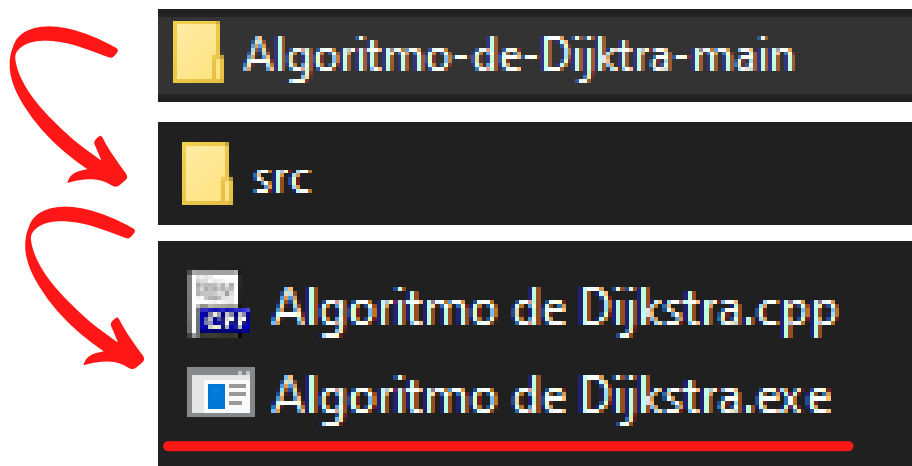
3. En el botón desplegable de "Code" hay varias formas para copiar el repositorio. En esta ocasión se seleccionará la forma de "Descargar ZIP".



4. Descargar el proyecto y buscarlo en la carpeta en la que se descargó.



5. Extraer el archivo y buscar la carpeta src. Al entrar a la carpeta encontrará los siguientes archivos:



6. Seleccionar el archivo “Algoritmo de Dijkstra.exe” y abrirlo.

A screenshot of a terminal window titled 'C:\Users\lucia\Downloads\Algoritmo-de-Dijkstra-main\src\Algoritmo de Dijkstra.exe'. The terminal displays the following text:

```
| Algoritmo de Dijkstra en C++ |  
-----  
1. Encontrar el camino mas corto  
2. Matriz Adyacencia  
3. Salir  
Ingrese una opcion:
```


7. Al abrir el proyecto mostrará la pantalla anterior y enseñará un menú. El menú cuenta con tres opciones principales.

```
-----  
|  Algoritmo de Dijkstra en C++  |  
-----  
  
1. Encontrar el camino mas corto  
2. Matriz Adyacencia  
3. Salir  
  
Ingrese una opcion:
```

8. Si se selecciona la primera opción mostrará la siguiente pantalla. Esta opción permite encontrar el camino más corto para llegar a un vértice específico.

```
-----  
--ENCONTRAR EL CAMINO MAS CORTO--  
-----  
Ingrese Cantidad de Vertices:
```

9. Primero se ingresa la cantidad de vértices y aristas que se desee.

```
Ingrese Cantidad de Vertices: 5  
Ingrese Cantidad de Aristas: 8
```

10. Luego de ingresar los valores, pedirá automáticamente que se ingrese valor del nodo origen, nodo destino y el peso de la arista que une ambos nodos.

Las veces que se ingresen los datos dependerá de las aristas que se tenga.

```
0 - Ingrese Origen: 1
0 - Ingrese Destino: 2
0 - Ingrese Peso de la Arista: 6
-----
1 - Ingrese Origen: 2
1 - Ingrese Destino: 5
1 - Ingrese Peso de la Arista: 3
-----
2 - Ingrese Origen: 3
2 - Ingrese Destino: 4
2 - Ingrese Peso de la Arista: 6
-----
3 - Ingrese Origen: 4
3 - Ingrese Destino: 5
3 - Ingrese Peso de la Arista: 2
-----
4 - Ingrese Origen: 1
4 - Ingrese Destino: 3
4 - Ingrese Peso de la Arista: 5
-----
5 - Ingrese Origen: 3
5 - Ingrese Destino: 2
5 - Ingrese Peso de la Arista: 4
-----
6 - Ingrese Origen: 1
6 - Ingrese Destino: 4
6 - Ingrese Peso de la Arista: 8
-----
7 - Ingrese Origen: 2
7 - Ingrese Destino: 4
7 - Ingrese Peso de la Arista: 5
```

11. Para finalizar pide ingresar el vértice inicial y el final, para buscar el recorrido.

12. Después de ingresar los valores mostrará la solución del algoritmo.

```
Ingrese Vertice Inicial: 2
Ingrese Vertice Final: 4
-----
Longitud del Camino mas Corto: 5
-----
```

13. Después de mostrar el resultado, regresará automáticamente al menú principal.

14. Ahora se seleccionará la segunda opción.

```
-----  
|  Algoritmo de Dijkstra en C++  |  
-----  
  
1. Encontrar el camino mas corto  
2. Matriz Adyacencia  
3. Salir  
  
Ingresa una opcion: 2 ←
```

15. Esto llevará a la siguiente pantalla y mostrará la matriz adyacencia.

```
-----  
-----MATRIZ ADYACENCIA-----  
-----  
          1      2      3      4      5  
1          0      1      0      1      0  
2          1      0      1      0      1  
3          0      1      0      1      0  
4          1      0      1      0      1  
5          0      1      0      1      0  
  
Presione una tecla para continuar . . .
```

16. Los datos para encontrar la matriz adyacencia se pueden modificar en el código fuente del programa.

17. Abrir el archivo "Algoritmo de Dijkstra.cpp" y dirigirse al "void menu". Esto se encuentra en las últimas líneas de código del programa.

```

335 void menu(){
336     system("cls");
337     int op1=0,valor=0; //creamos 2 variables enteras.
338     lista l; //creamos una nueva clase.
339     Programa programa; //TAD
340     Graph graph; // Grafo.
341
342     l.Insertarv(1); ← a.
343     l.Insertarv(2);
344     l.Insertarv(3); //creamos y asignamos valores
345     l.Insertarv(4);
346     l.Insertarv(5);
347     l.creararco(1,2,1);
348     l.creararco(2,1,1);
349     l.creararco(1,4,2);
350     l.creararco(4,1,2);
351     l.creararco(4,3,3);
352     l.creararco(3,4,3); ← b.
353     l.creararco(3,2,4);
354     l.creararco(2,3,4);
355     l.creararco(4,5,5);
356     l.creararco(5,4,5);
357     l.creararco(5,2,6);
358     l.creararco(2,5,6);
359 }

```

a. Sirve para crear los vértices.

b. Sirve para poner la relación entre los vértices.

18. El primero y segundo número entre los paréntesis sirve para colocar las aristas que están relacionadas y en el tercer número se coloca el nombre de la arista.

```

l.creararco(1,2,1);

```

19. Luego de esto se guardan los cambios y se vuelve a ejecutar. Se mostrarán los cambios realizados.

20. La tercera opción del menú se utiliza para salir de programa. Después de seleccionarlo llevará a la siguiente pantalla. Y se terminará la ejecución del programa

```

GRACIAS POR PROBAR!
-----
Process exited after 8.795 seconds with return value 0
Presione una tecla para continuar . . .

```