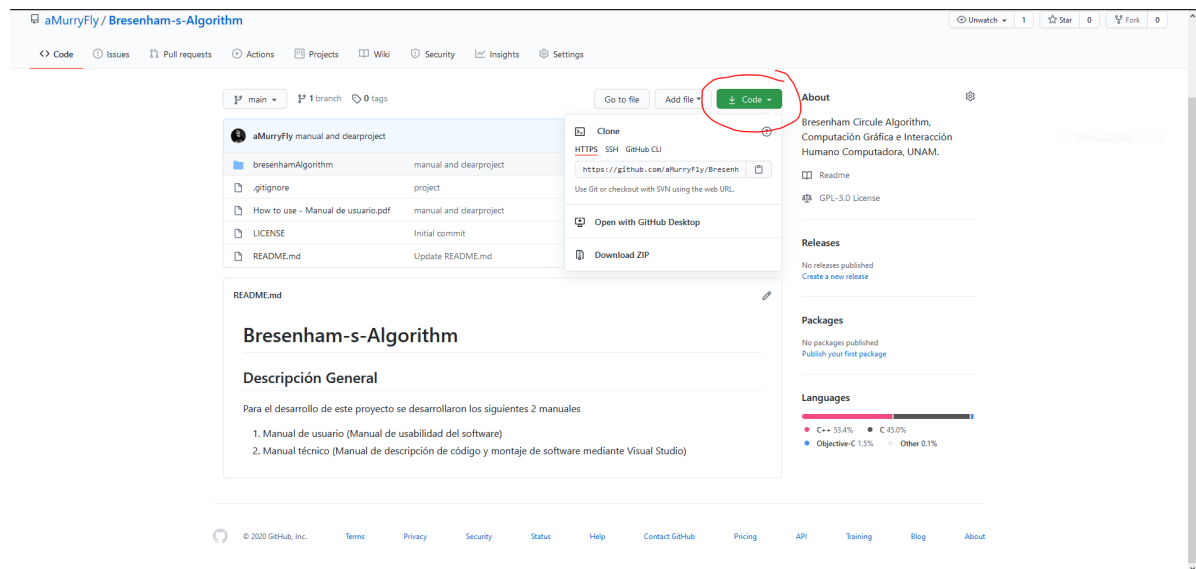


# How to compile (Manual Técnico)

## 1) Descargar

Para poder compilar y probar el proyecto, es necesario descargar el proyecto de Github (Es necesario destacar que el proyecto está totalmente limpio de basura o de archivos generados por el IDE)

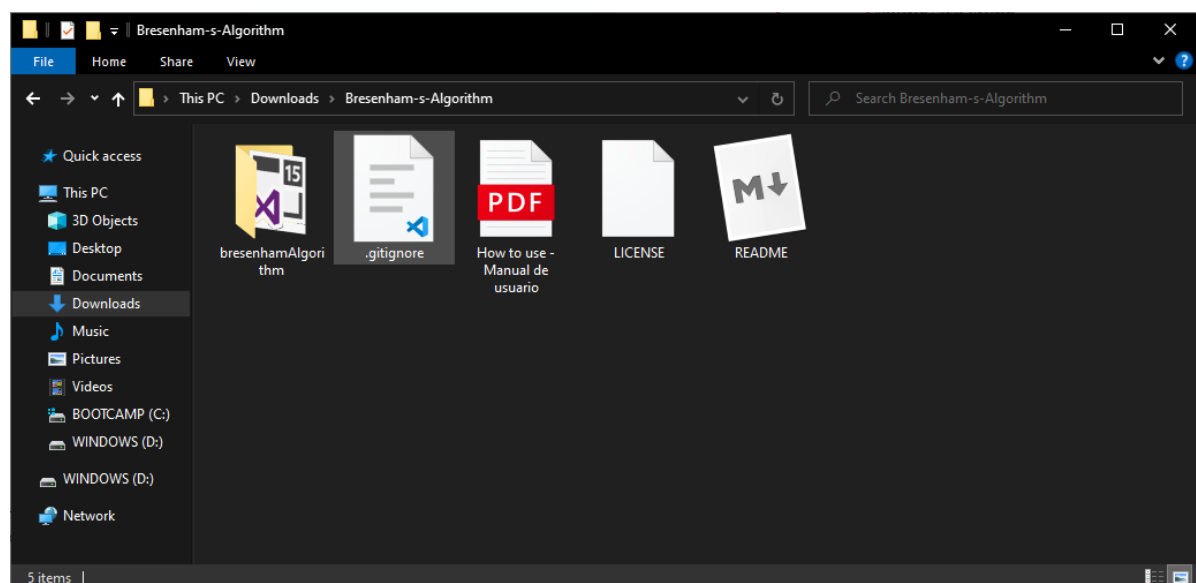


Para descargar puede directamente descargarse un archivozip del proyecto o en terminal colocar lo siguiente:

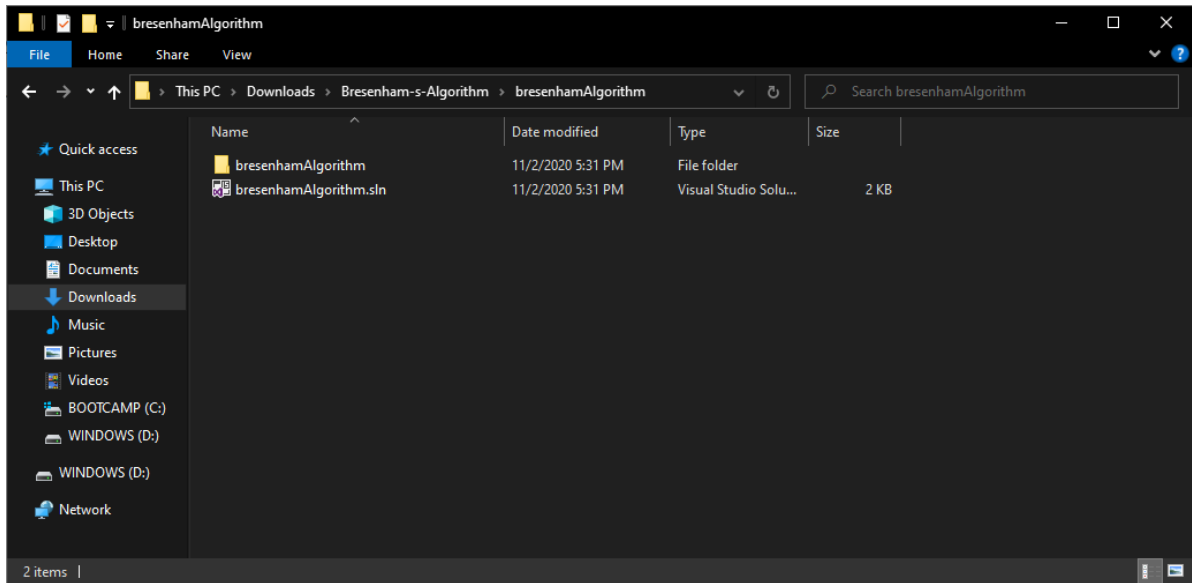
```
git clone https://github.com/aMurryFly/Bresenham-s-Algorithm.git
```

## 2) Montar en Visual Studio

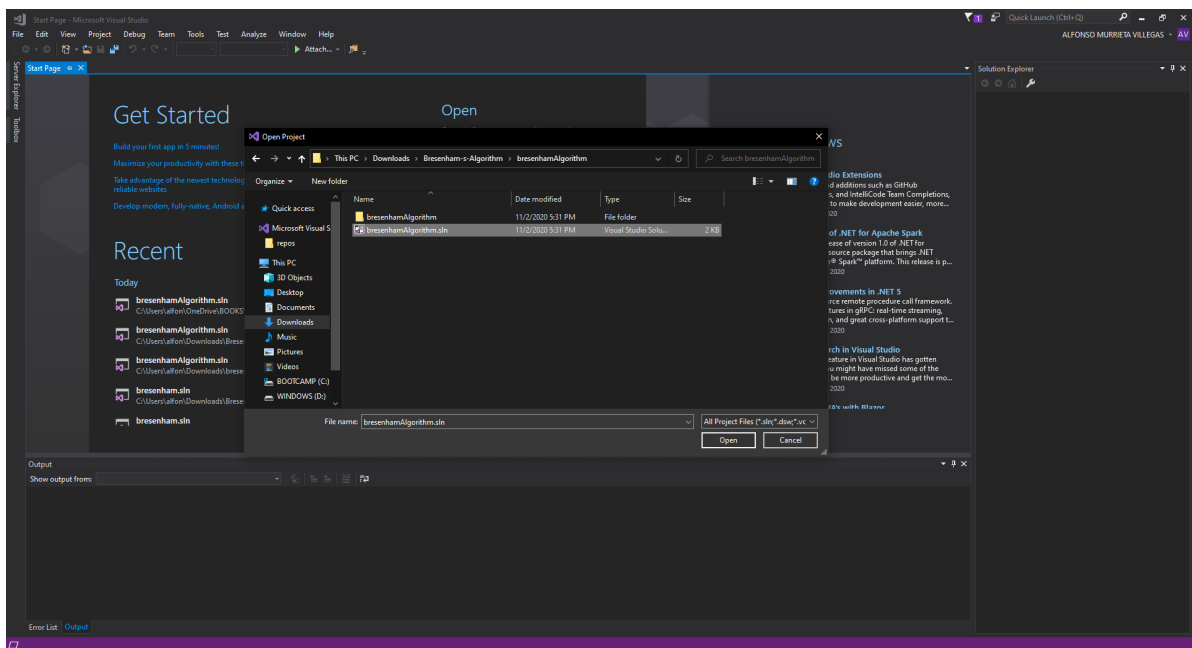
Una vez descargado acceder a la carpeta generada y se tendrá la siguiente estructura



Partiendo de la captura anterior se pueden hacer dos cosas, una es abrir el archivo *bresenhamAlgorithm.sln* que es el archivo para abrir en Visual Studio Community 2017

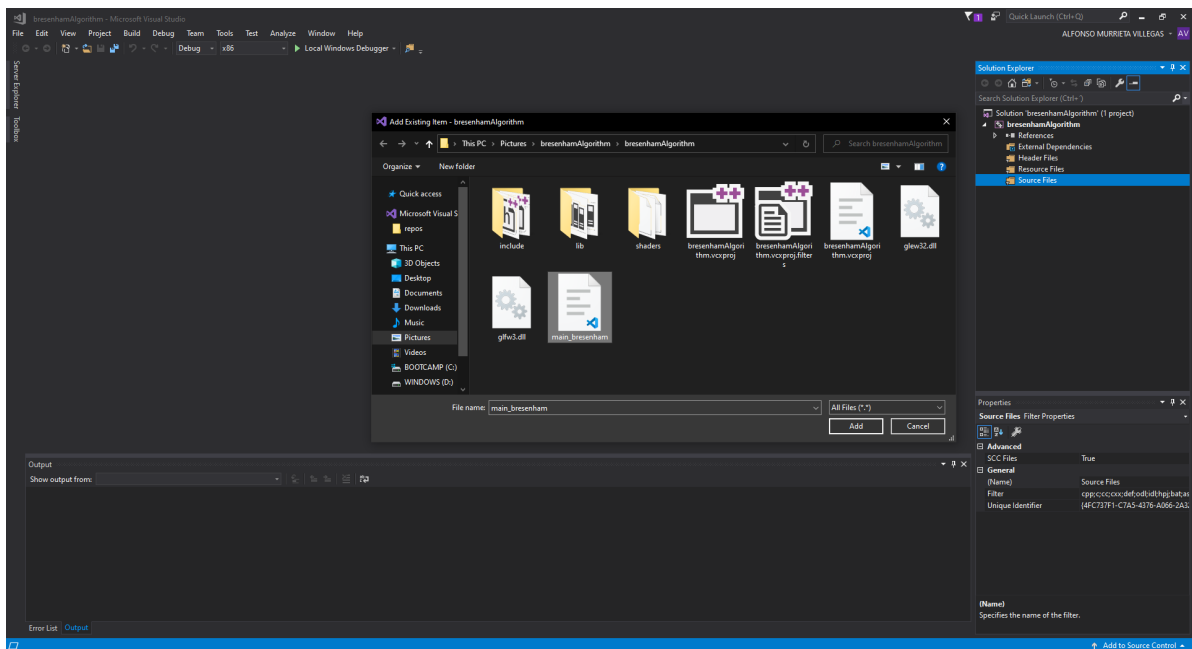


O también se puede abrir desde el mismo IDE (File>Open>Project), dar click sobre el archivo .sln y abrir

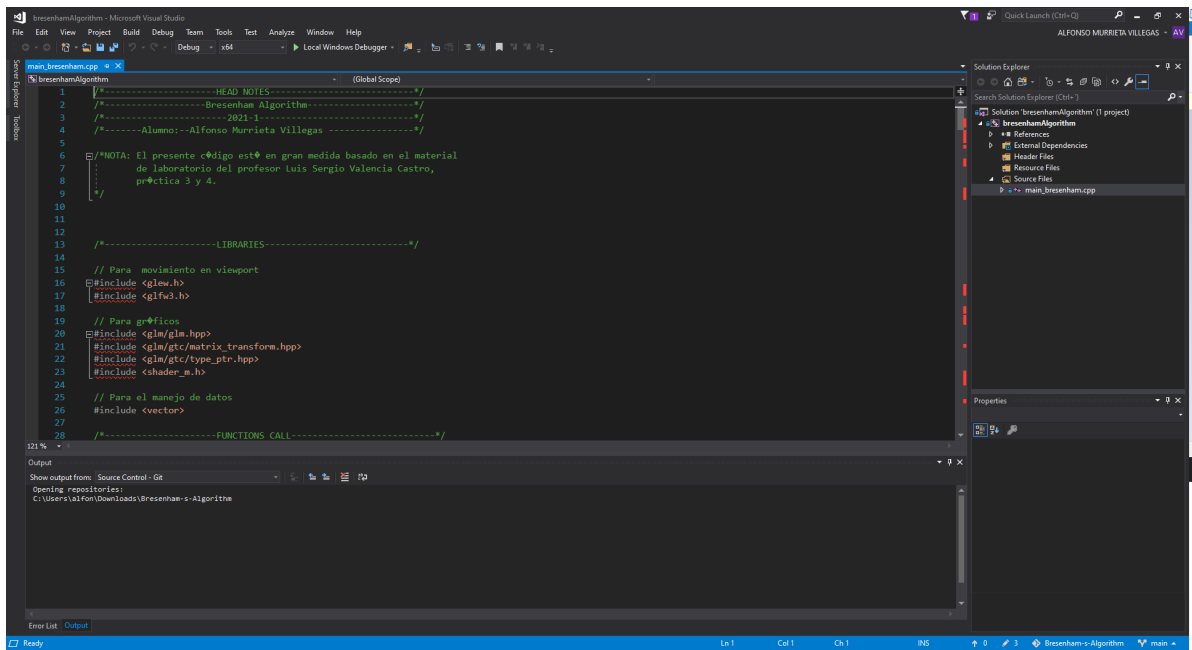


### 3) Configurar el proyecto para su ejecución

Importar el archivo main\_bresenham dentro de la carpeta del proyecto Source File



Como resultado debería verse el archivo main dentro de la carpeta *Source files* o *archivos de origen*



Como se observar en la imagen anterior se puede notar como es que nos manda el mismo IDE unas marcas rojas de que el archivo no se ha configurado del todo, esto principalmente a que falta importar las bibliotecas necesarias para correr el proyecto.

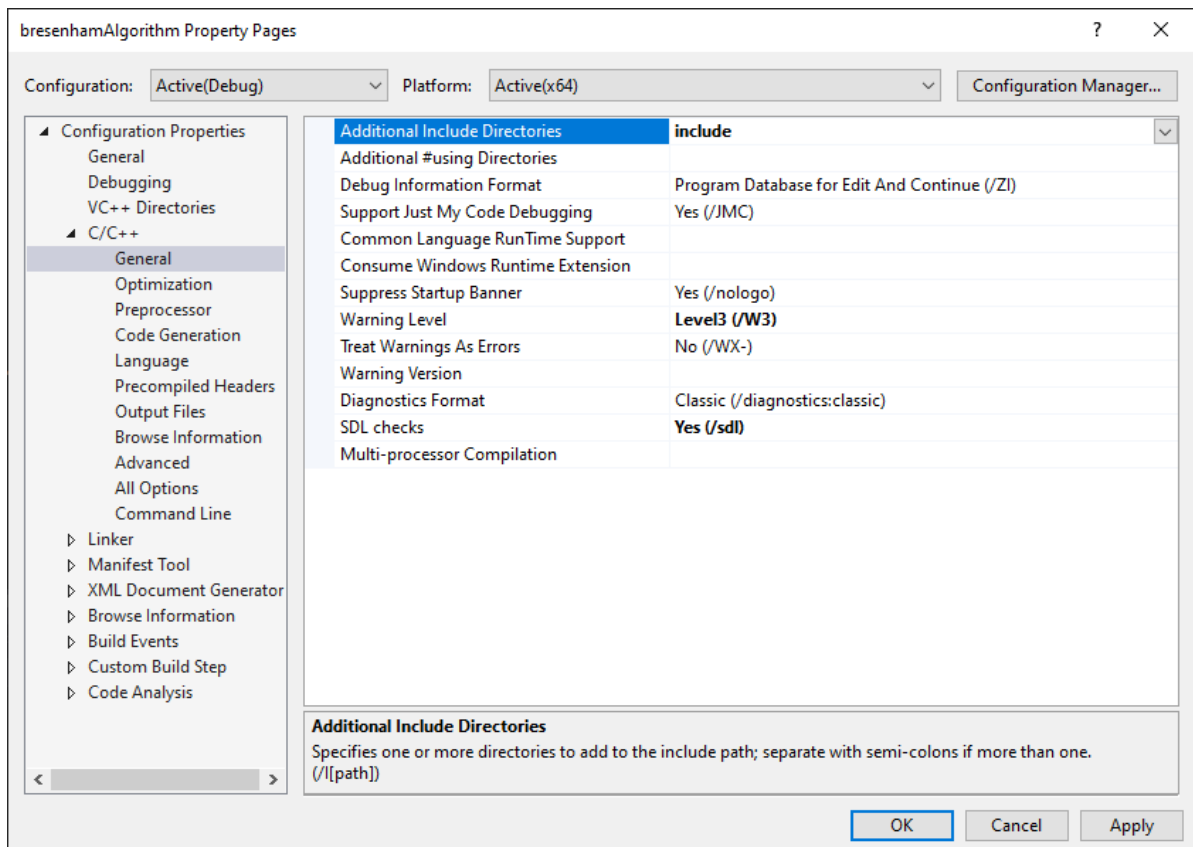
**NOTA:** Para este paso todo está basado en el "tutorial / manual " del profesor Sergio Valencia el cual a su vez se agregará en este repo,

Dar click derecho sobre el proyecto o solución y acceder al menú de propiedades

 Screenshot (3)

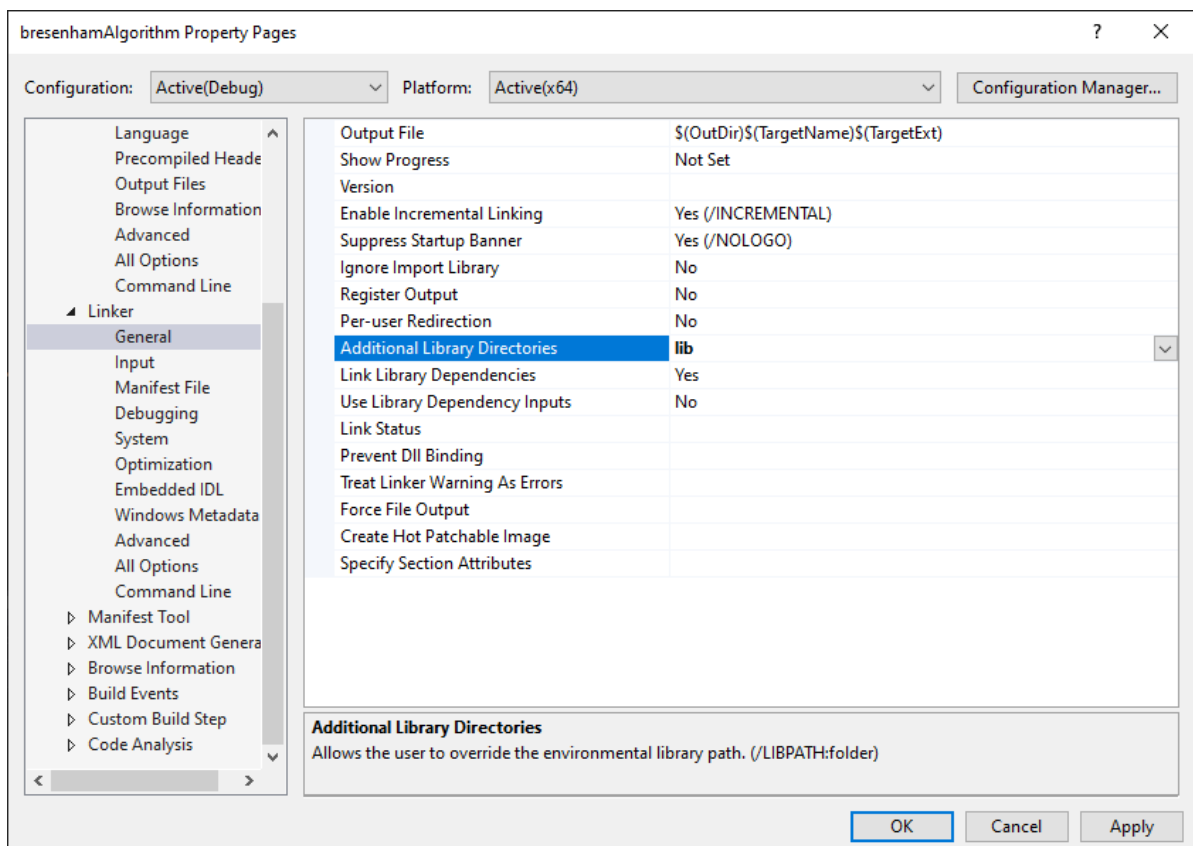
Dentro de la pantalla de menú agregar las siguiente bibliotecas y entradas al proyecto:

- **c/c++ > general > additional include**
  - Agregar en el campo *include*



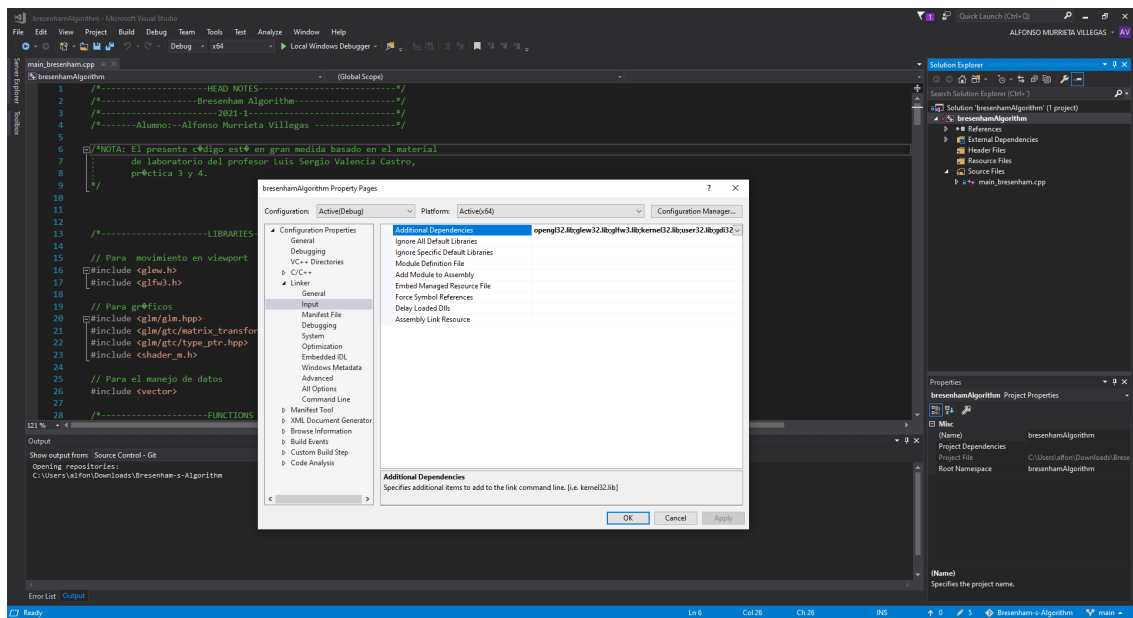
- **Linker > general > Additional library**

- Agregar *lib*



- **Linker>Input> Additional dependencies**

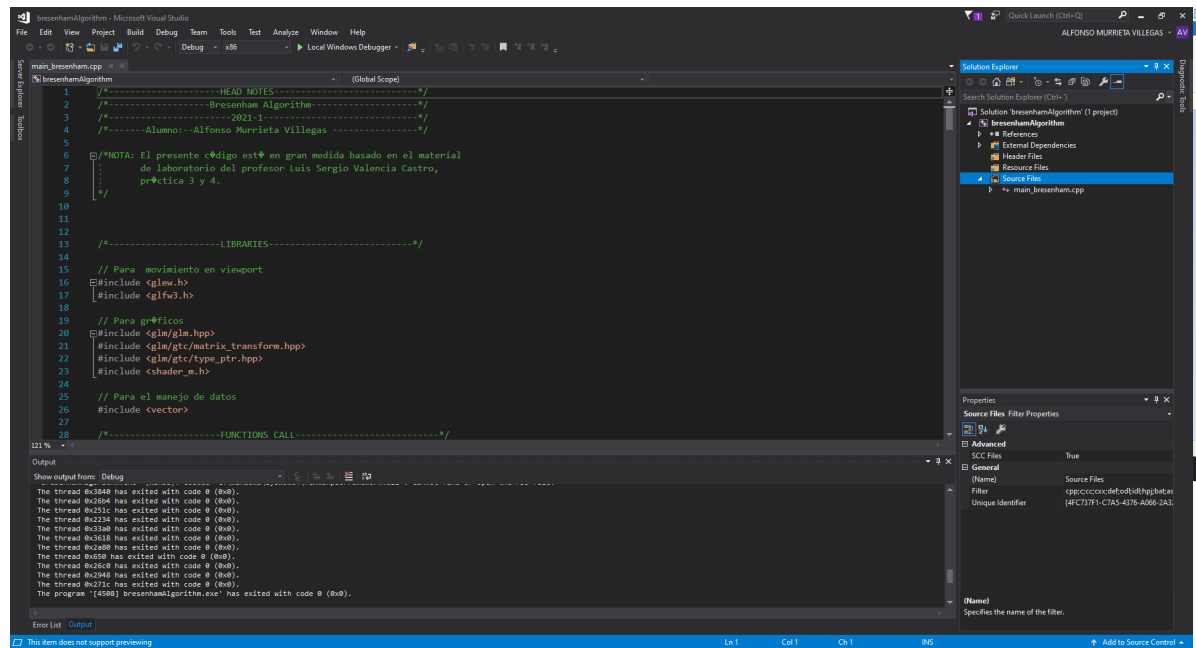
- Agregar *opengl32.lib;glew32.lib;glfw3.lib;*
- Una vez agregado, dar en aplicar y ok



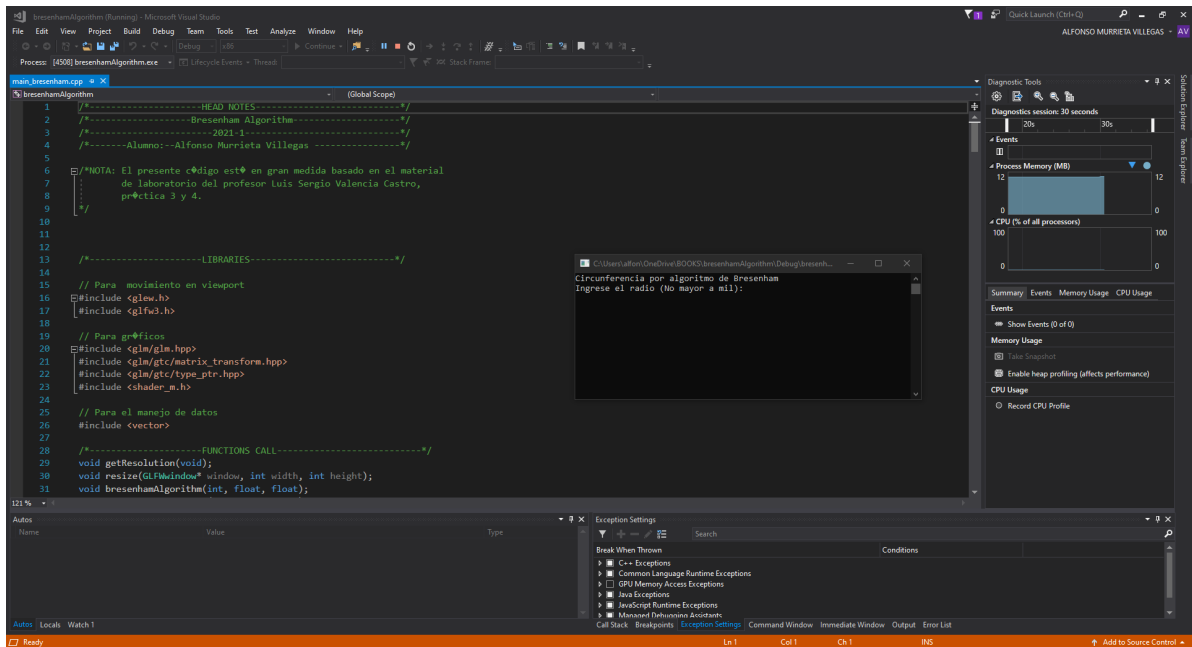
Se puede notar que las marcas rojas dentro del IDE han desaparecido, esto debido a que ya tiene todas las dependencias importadas correctamente

#### 4) Compilación y uso

Con base a lo anterior dar click en Local Windows Debugger o simplemente dar F5



Posterior a ello deben aparecer dos pantallas o ventanitas adicionales, una que es la pantalla o terminal negra denominada inputConsole



y por otro lado, debe a parecer otra ventana blanca denominada BresenhamAlgorithm



Para saber el manejo o forma de realizar las circunferencias, revisar el ***How to use o manual de usuario***

## EXTRA: En caso de tener problemas con el linker

Al inicio del proyecto, particularmente tuve problemas al importar de github mi proyecto otra vez en mi misma computadora, para ello existen dos soluciones:

**1 Sacar de la carpeta del repositorio de github la carpeta que tiene todo el proyecto de Visual Studio, suena extraño pero funciona.**

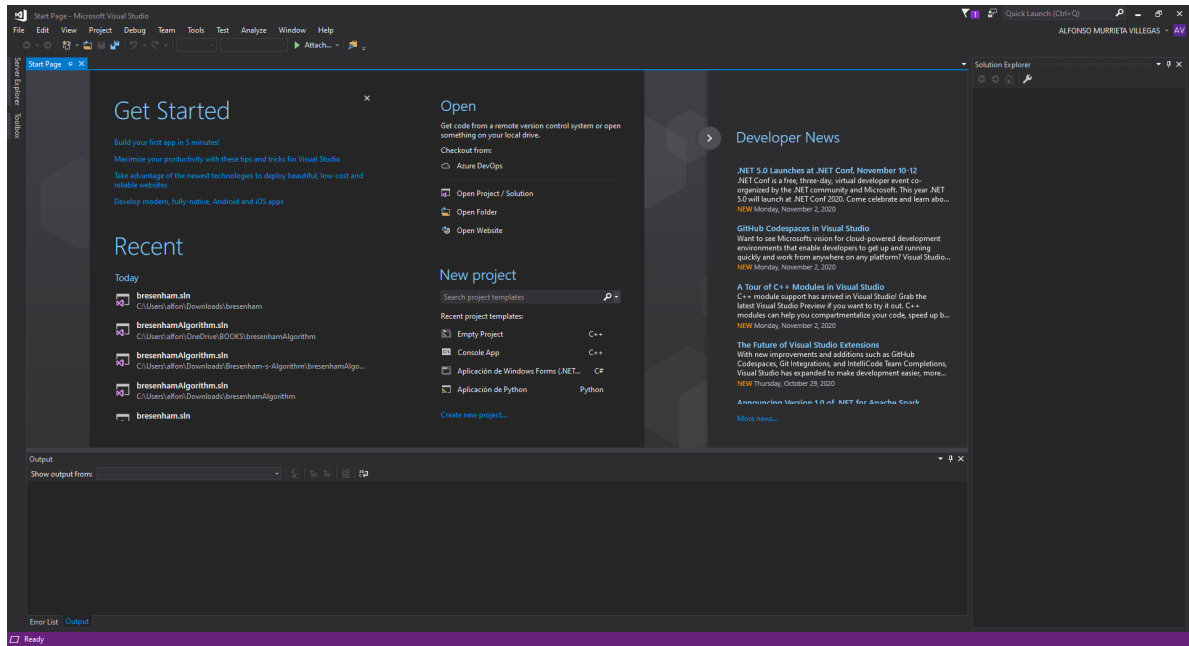
## 2 Crear un proyecto nuevo en Visual Studio

Suena bastante laborioso pero realmente no es tan difícil como puede aparentar.

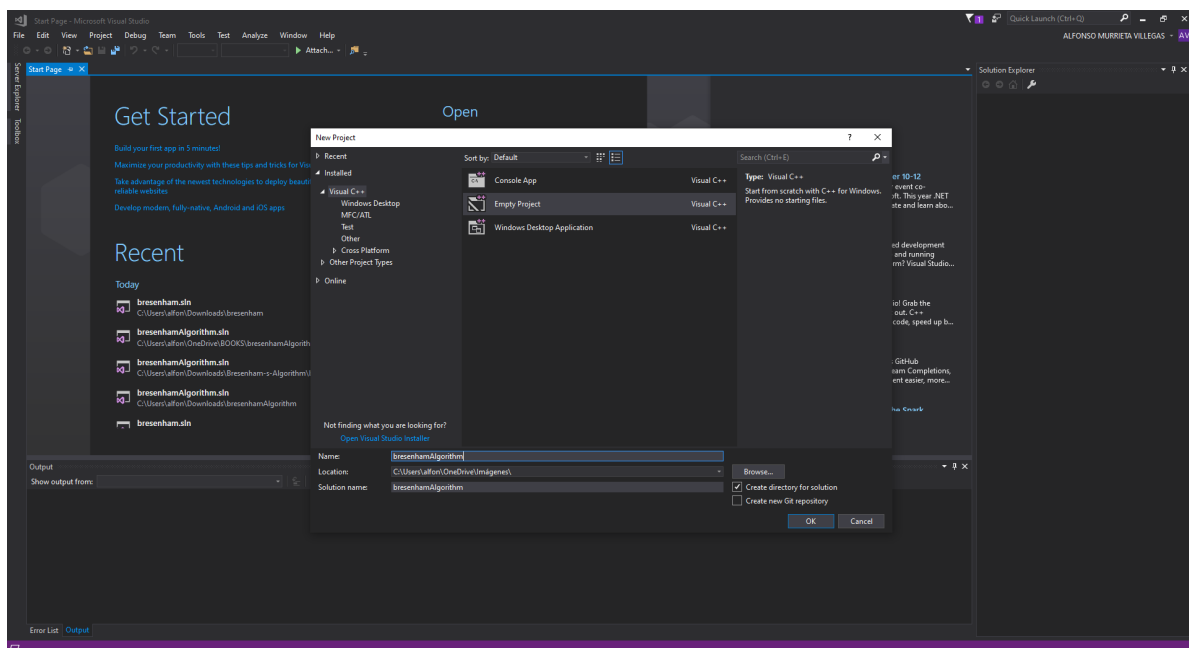
2.1) Dentro del mismo repositorio de Github o en el mismo Google Classroom, subiré un archivo rar denominado **resources.rar** (Descargarlo)

- Ese archivo incluye las bibliotecas, shaders y main code de mi proyecto

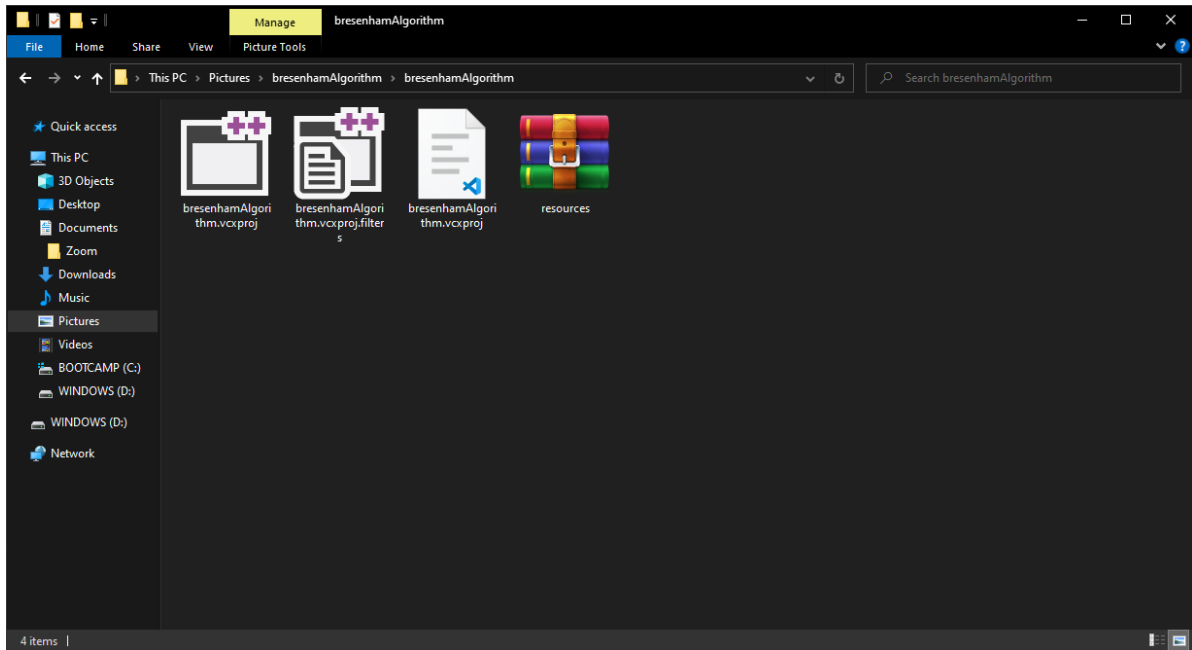
2.2) Creamos un nuevo proyecto en Visual Studio Community



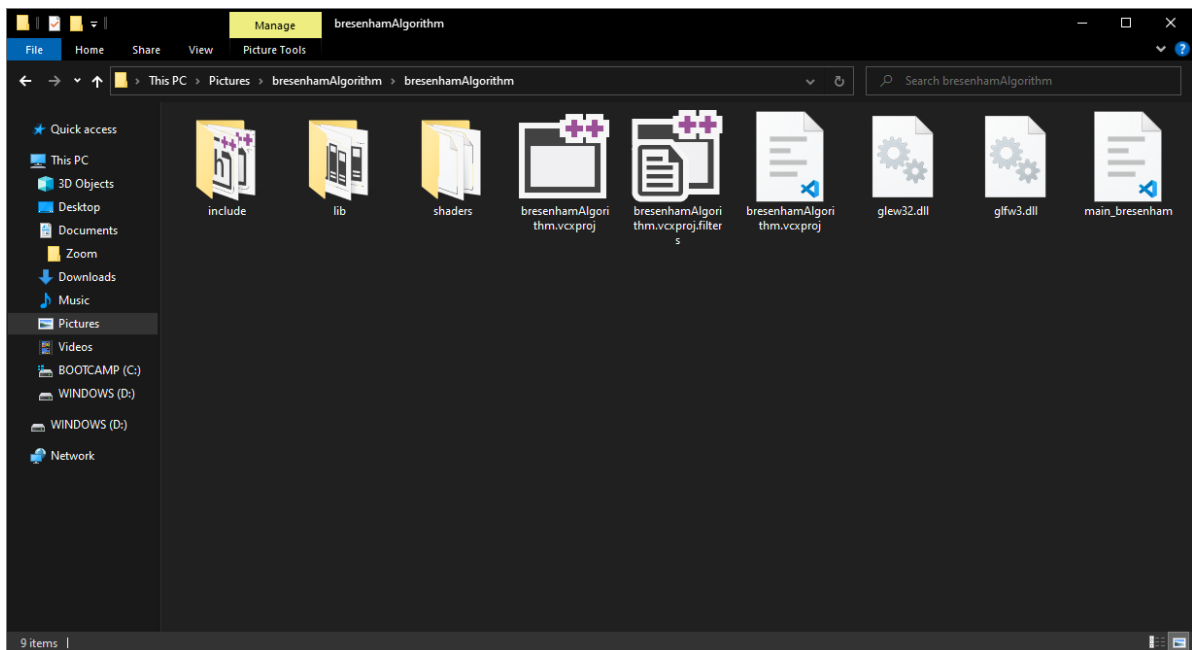
Lo creamos como proyecto vacío, escogemos la dirección donde queremos crear el proyecto y asignamos un nombre al proyecto



2.3) Movemos el archivo rar a la dirección donde se creó el proyecto, específicamente en donde se creó la carpeta del source code del proyecto.

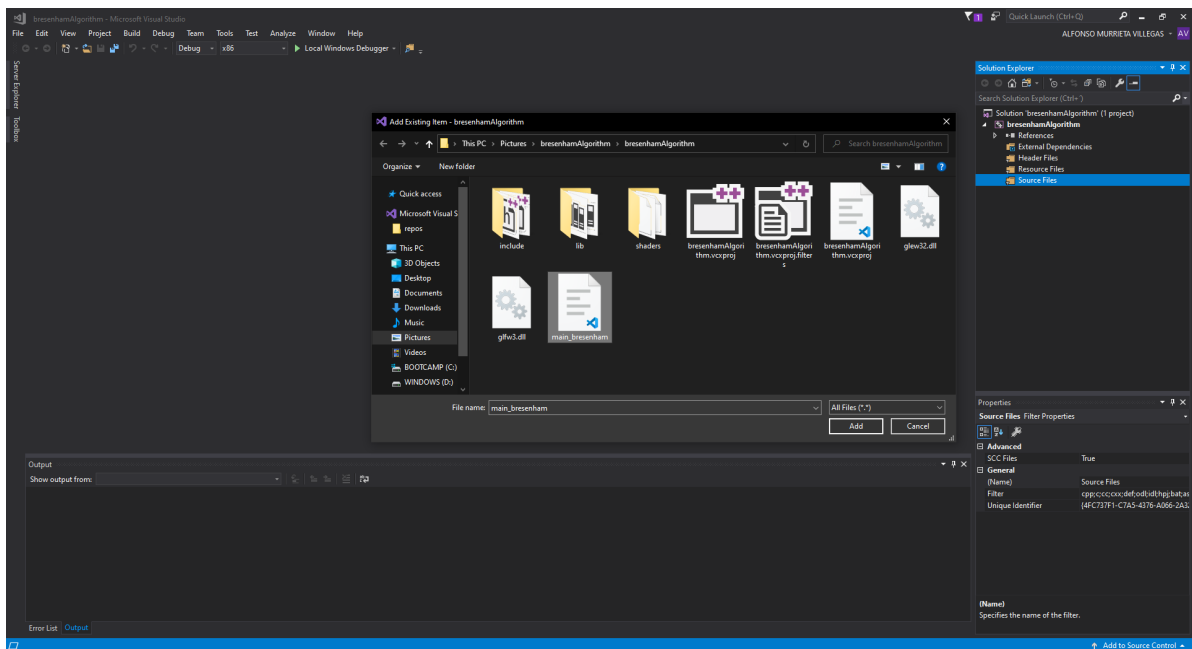


2.4) descomprimos el archivo y borramos el archivo rar

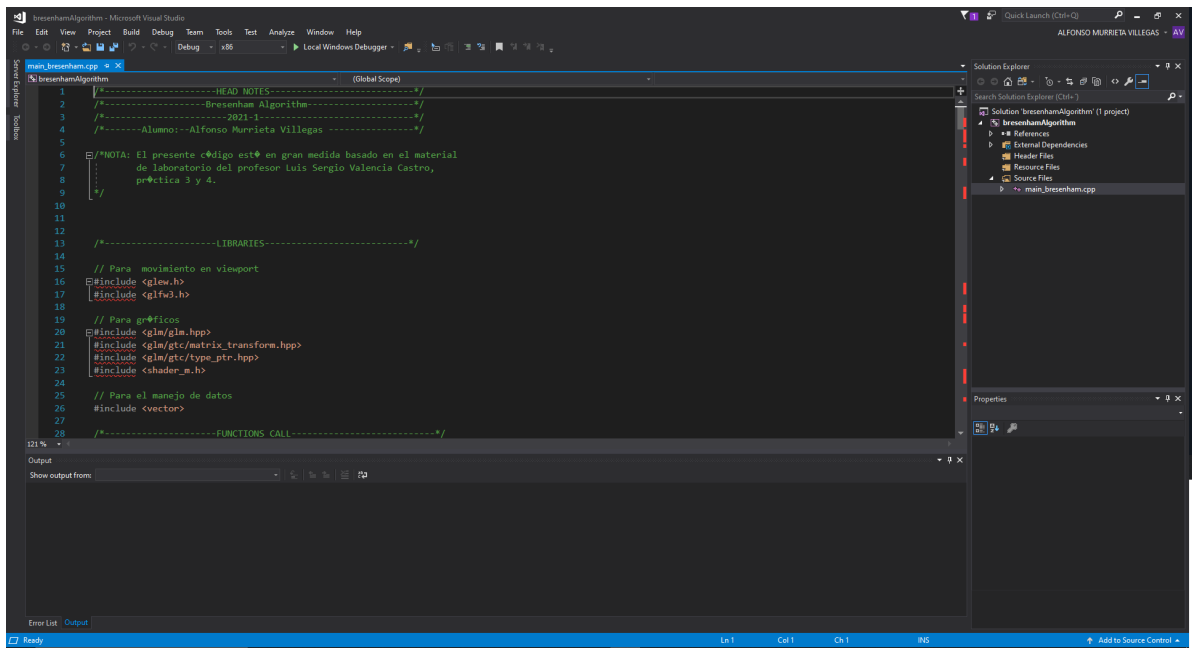


2.5 ) dentro de nuestro proyecto en Visual Studio agregamos el archivo "main\_bresenham.cpp", en el apartado de source files o archivos de origen





2.6) Como resultado obtendremos lo siguiente:



Como se puede apreciar estaríamos ahora desde el punto "**3) Configurar el proyecto para su ejecución**", es decir debemos agregar las bibliotecas y recursos al proyecto y posteriormente compilarlo (*Seguir las mismas instrucciones del How to use a partir del apartado 3) Configurar el proyecto para su ejecución*).