

[Volver](#)

Como instalar la libreria de PseudoCodeWrapper

Index

[Proyecto Tutorial_Psuedo](#)[Crear proyecto de CMake](#)

- [Visual Studio 2022 - Visual Studio Code](#)

[Bajar la libreria](#)[Instalar libreria dentro del proyecto](#)

- [Visual Studio 2022 - Visual Studio Code](#)
-

Proyecto Tutorial_Psuedo

Link hacia el proyecto de ejemplo  

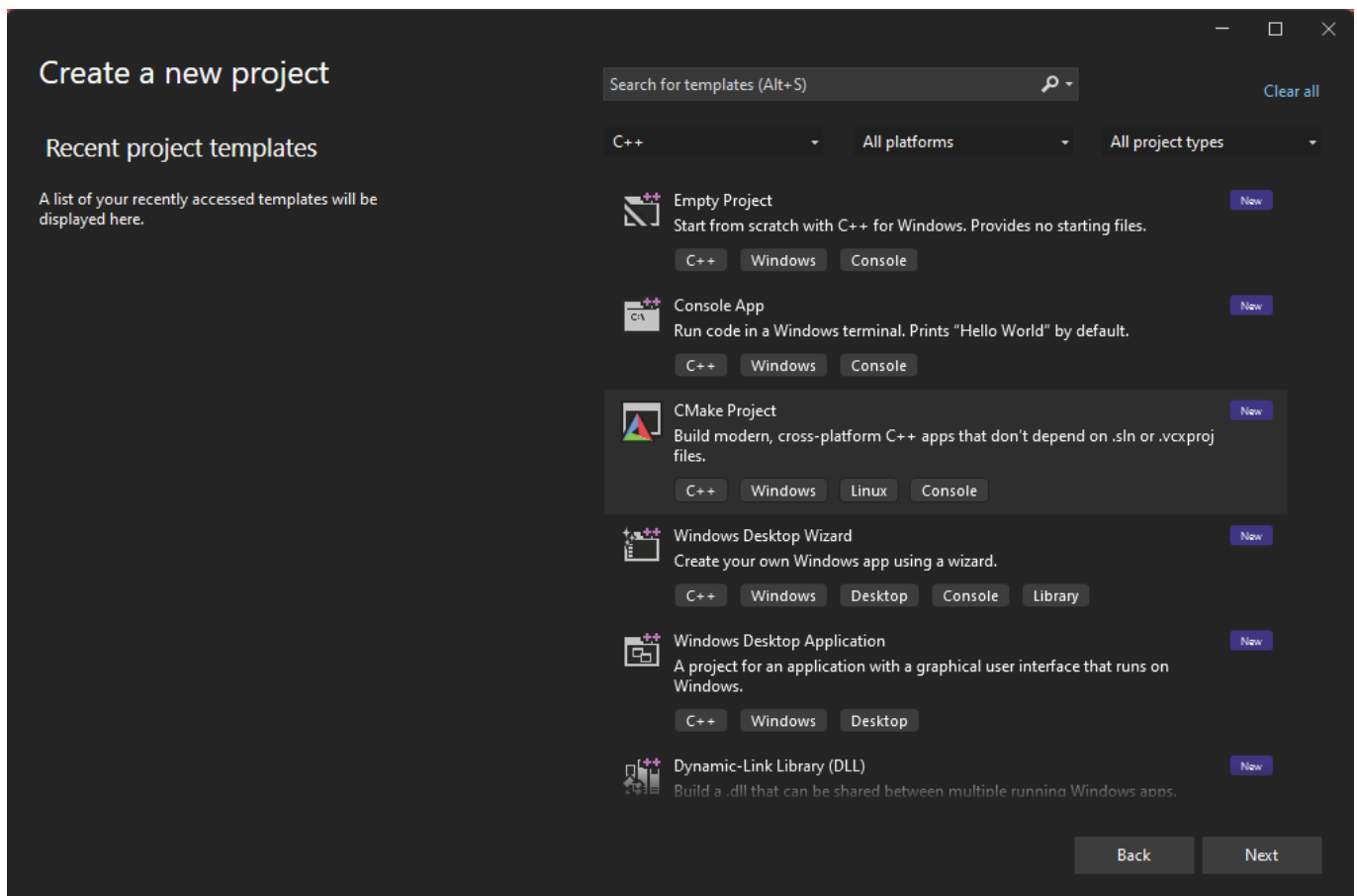
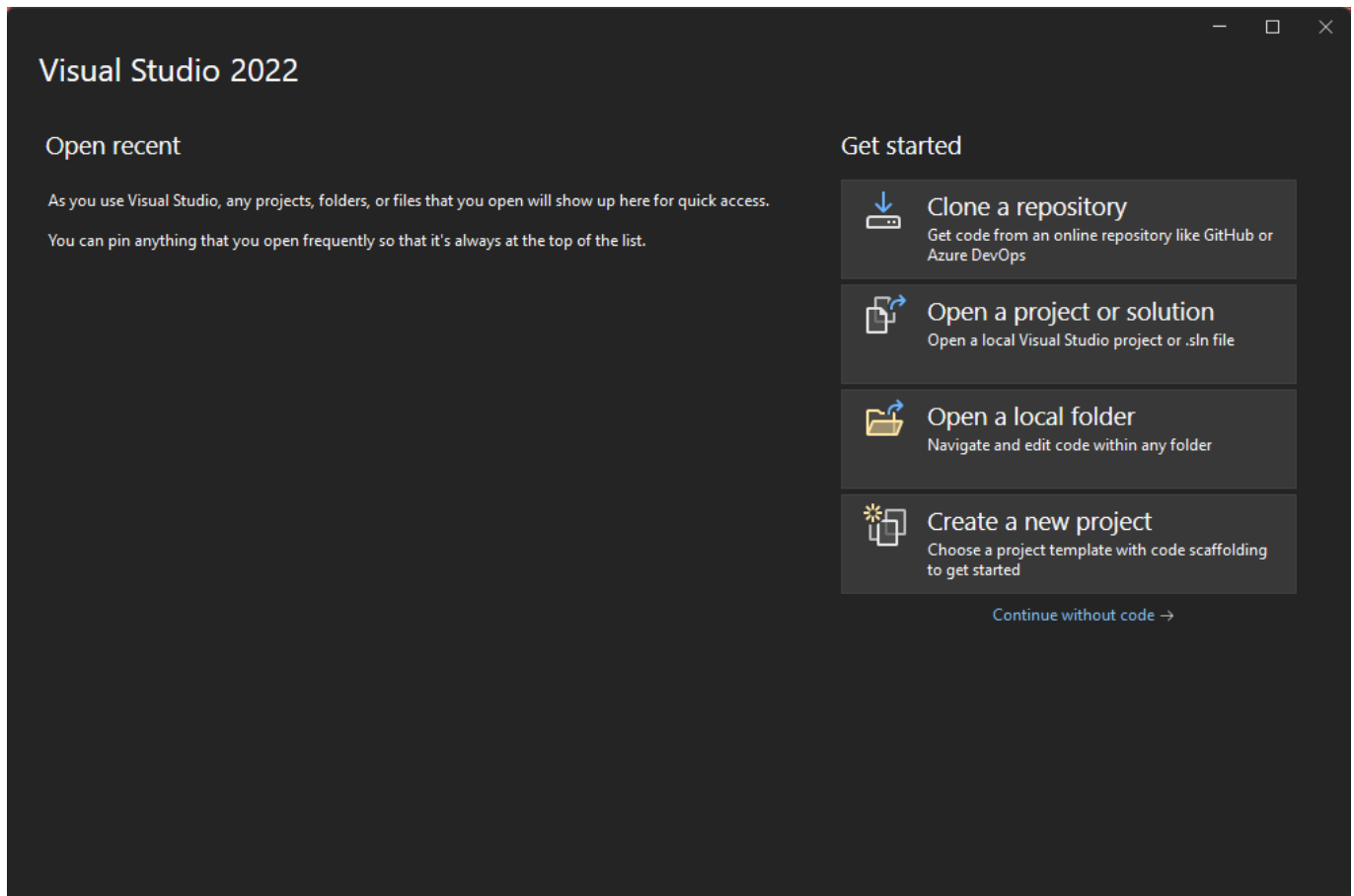
[Click Aqui](#)

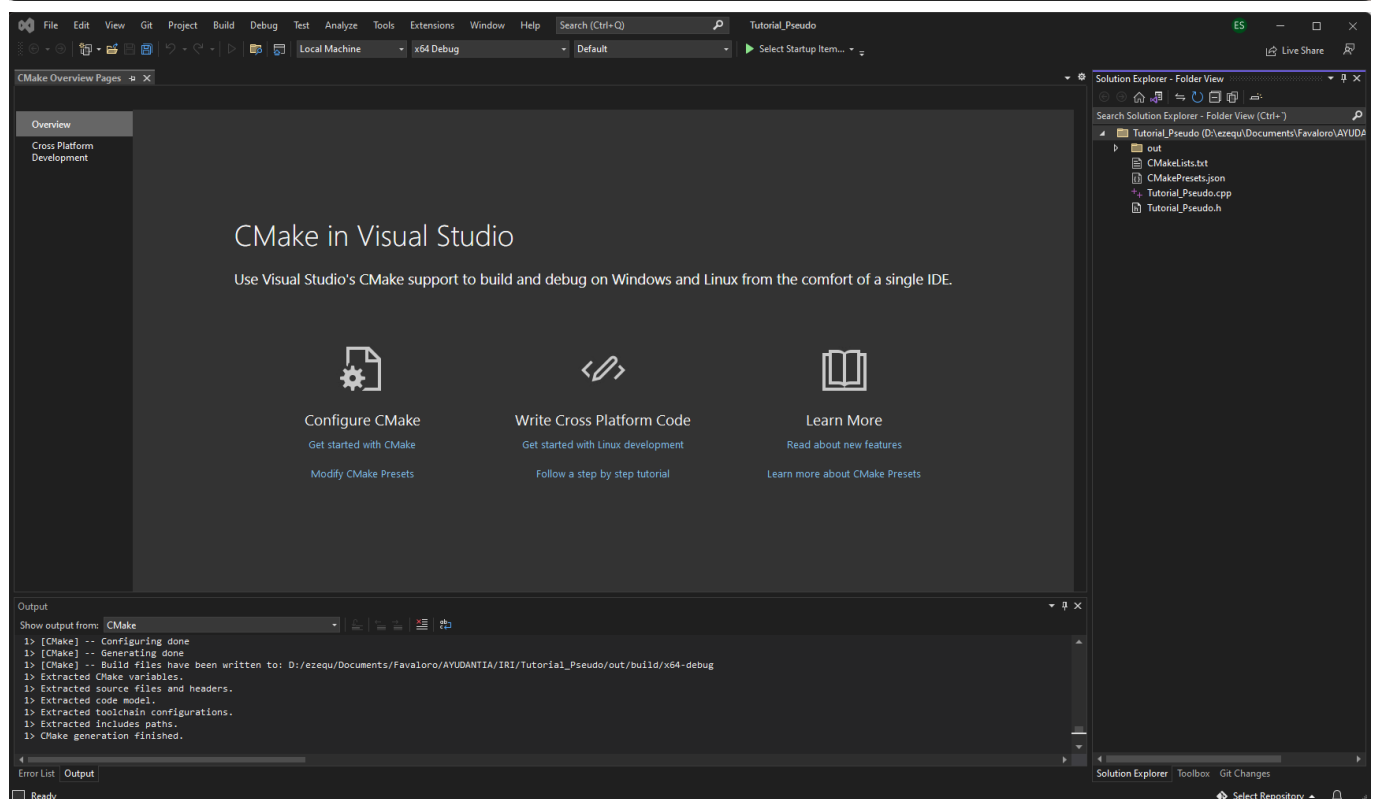
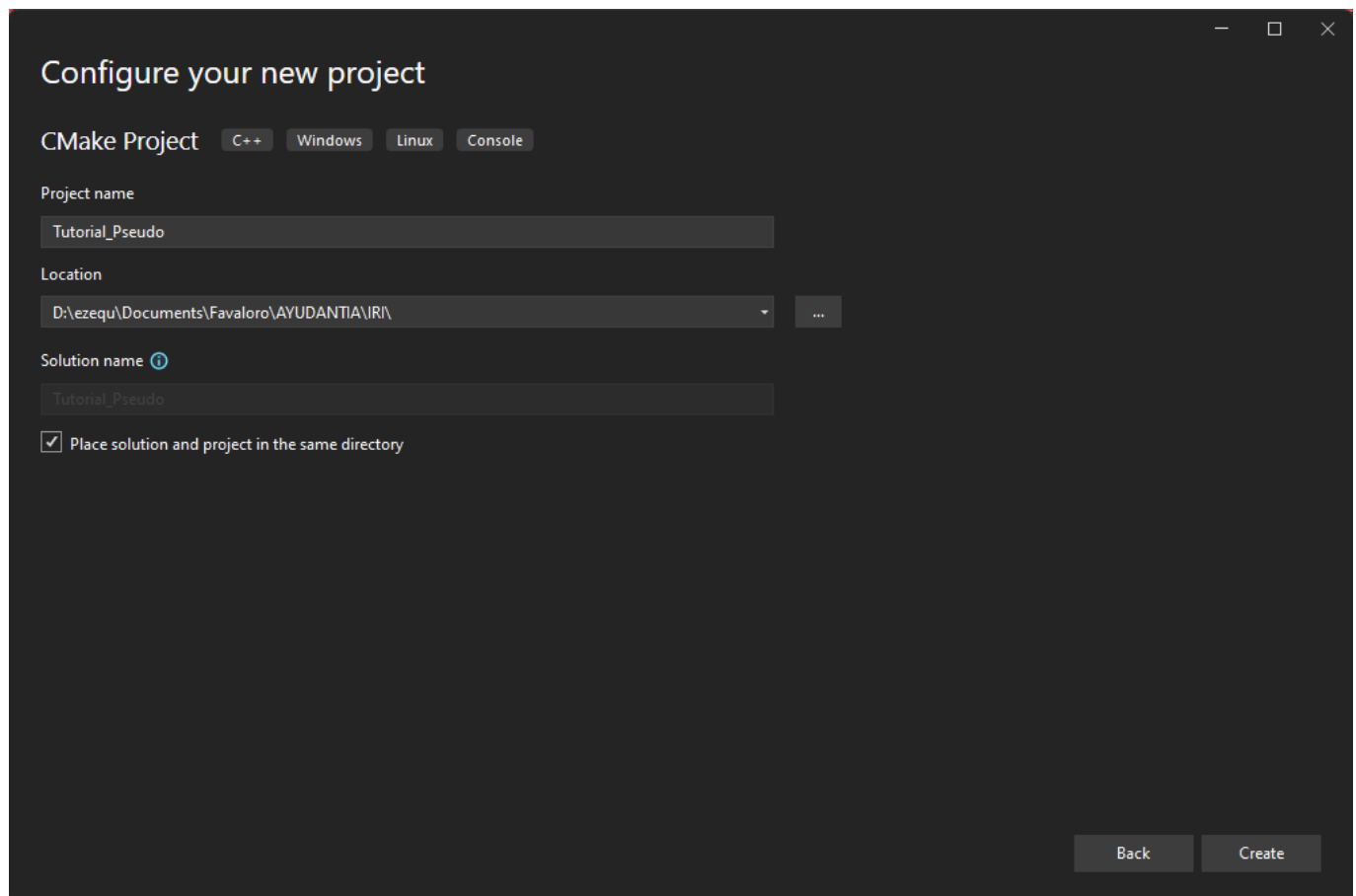
De esta forma deberia verse el proyecto bien creado

```
Tutorial_Pseudo
+-- CMakeLists.txt
+-- Tutorial_Pseudo.h
+-- Tutorial_Pseudo.cpp
+-- <Otros archivos>
|
+-- out/build
    +-- <Muchas cosas>
```

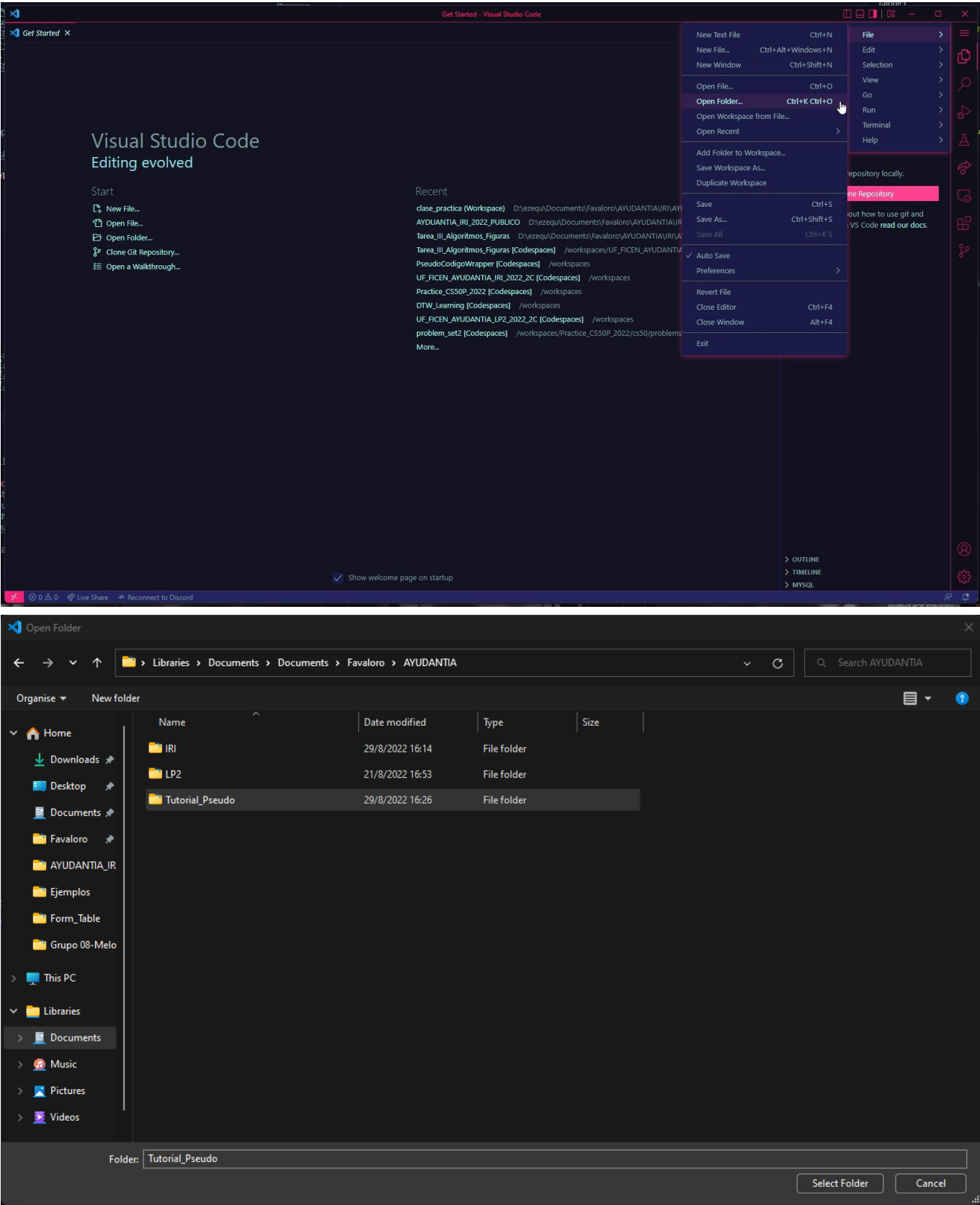
Crear proyecto de CMake

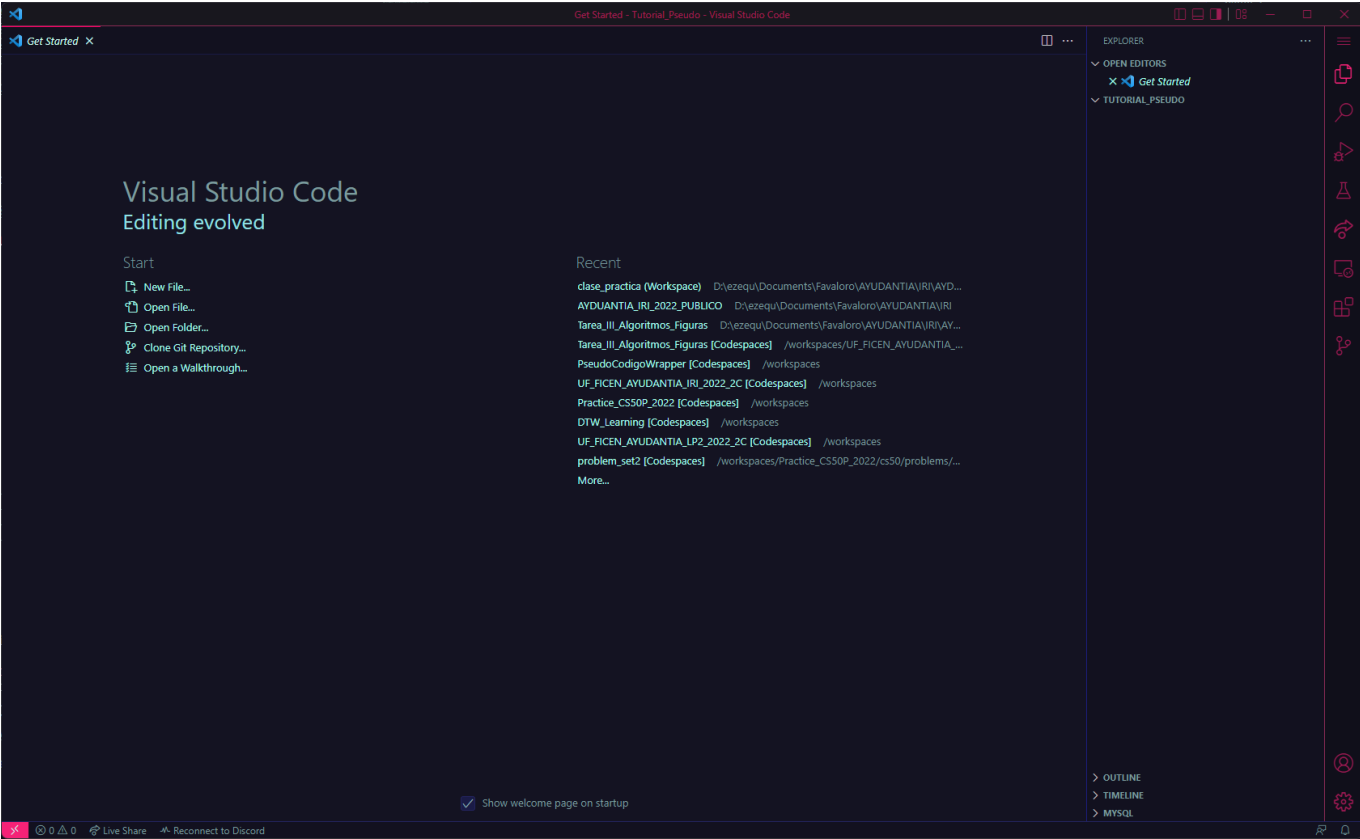
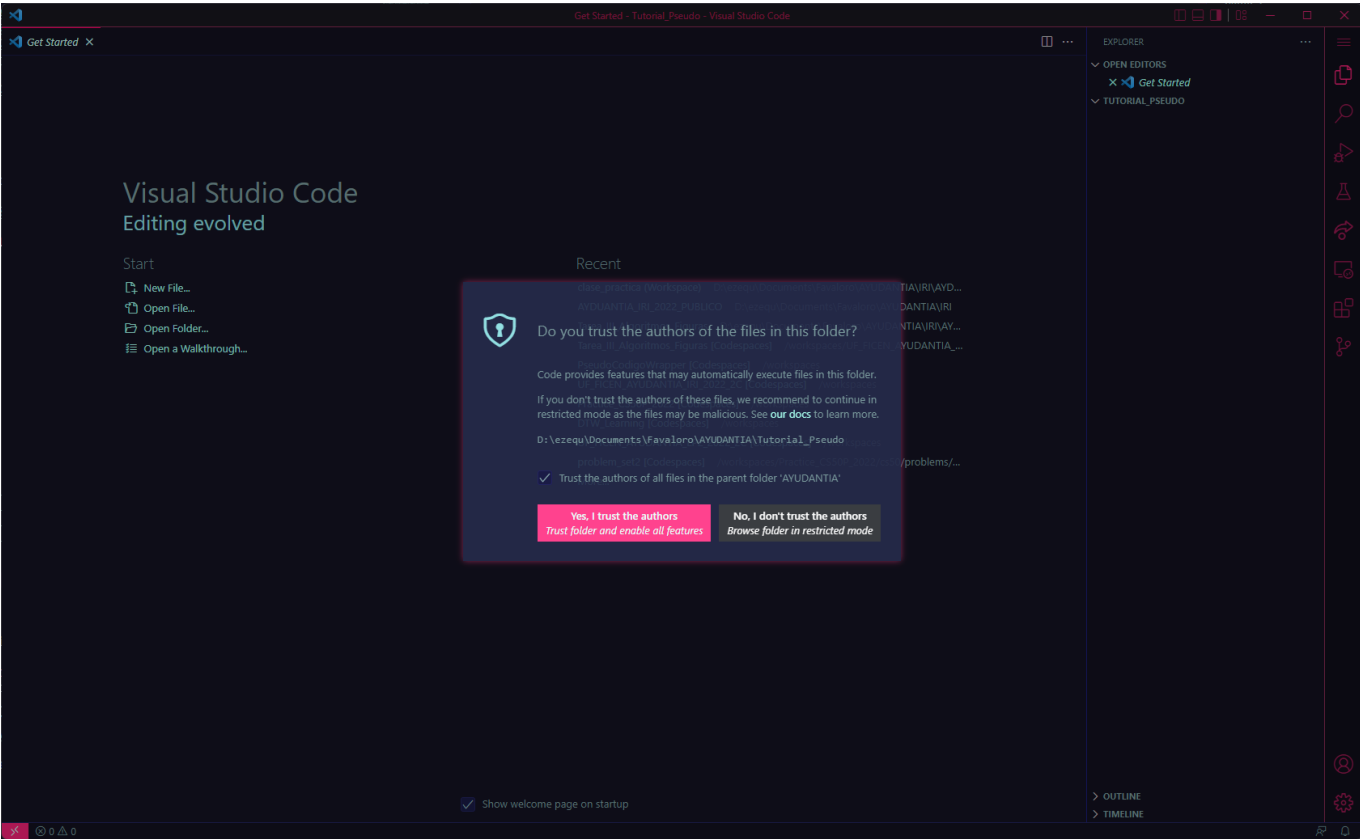
Visual Studio 2022

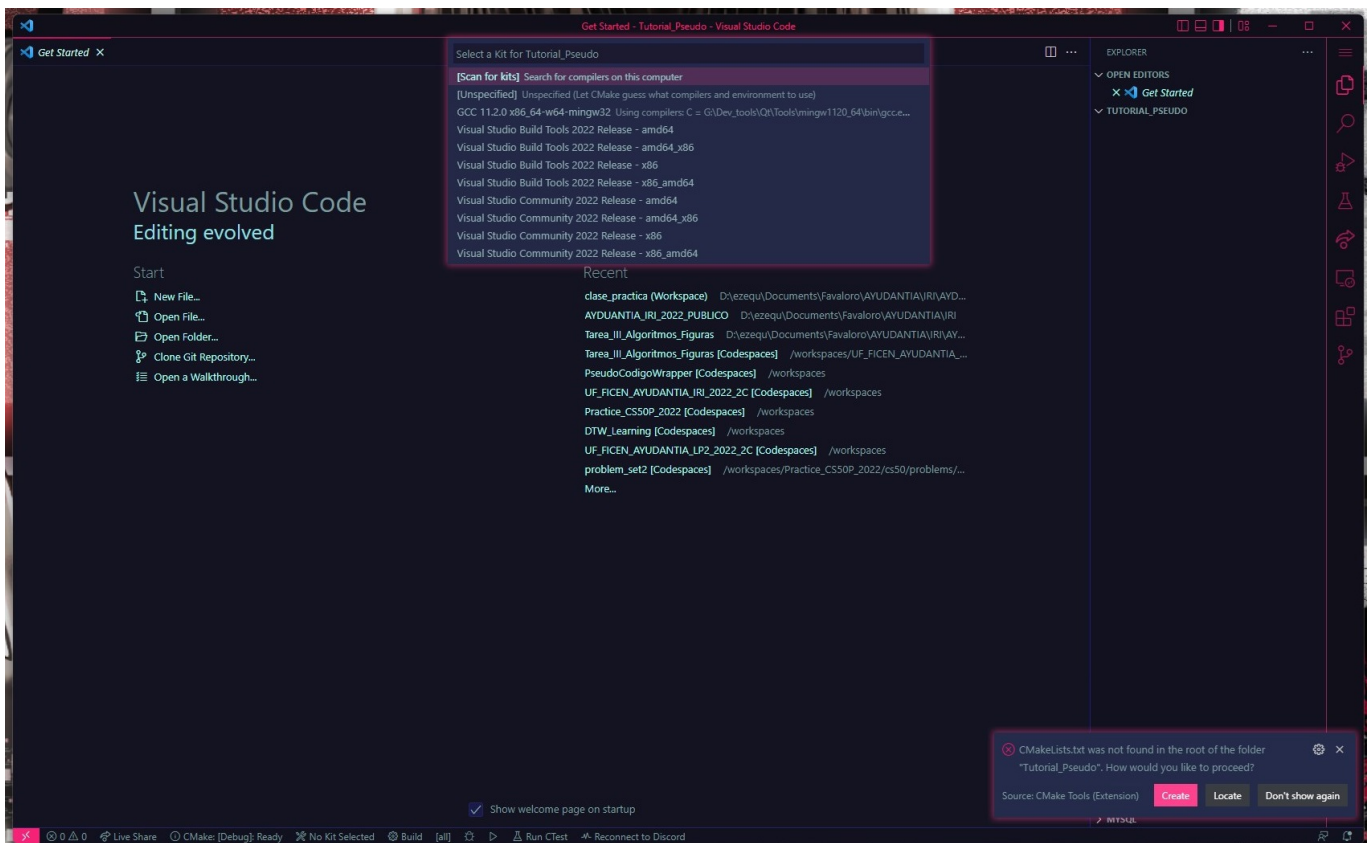
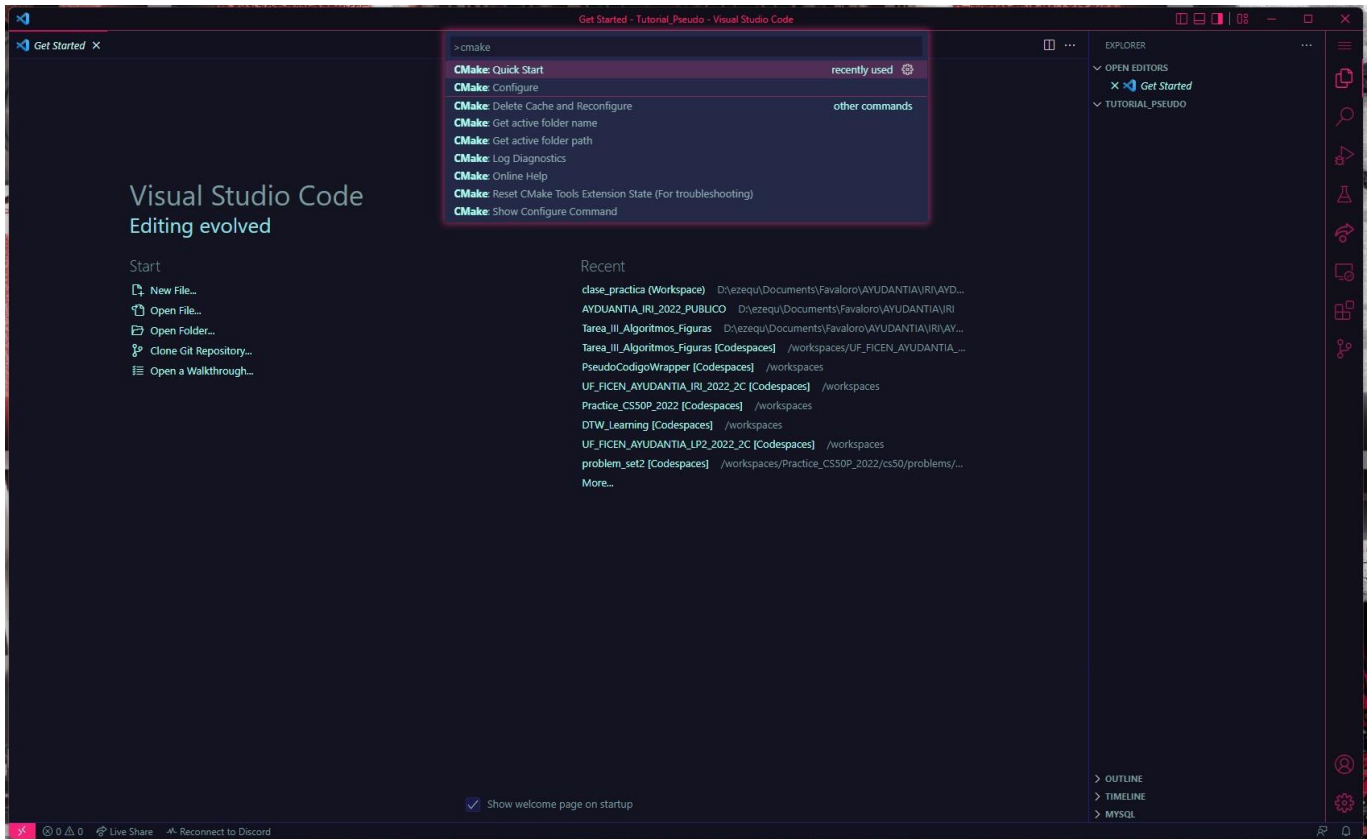


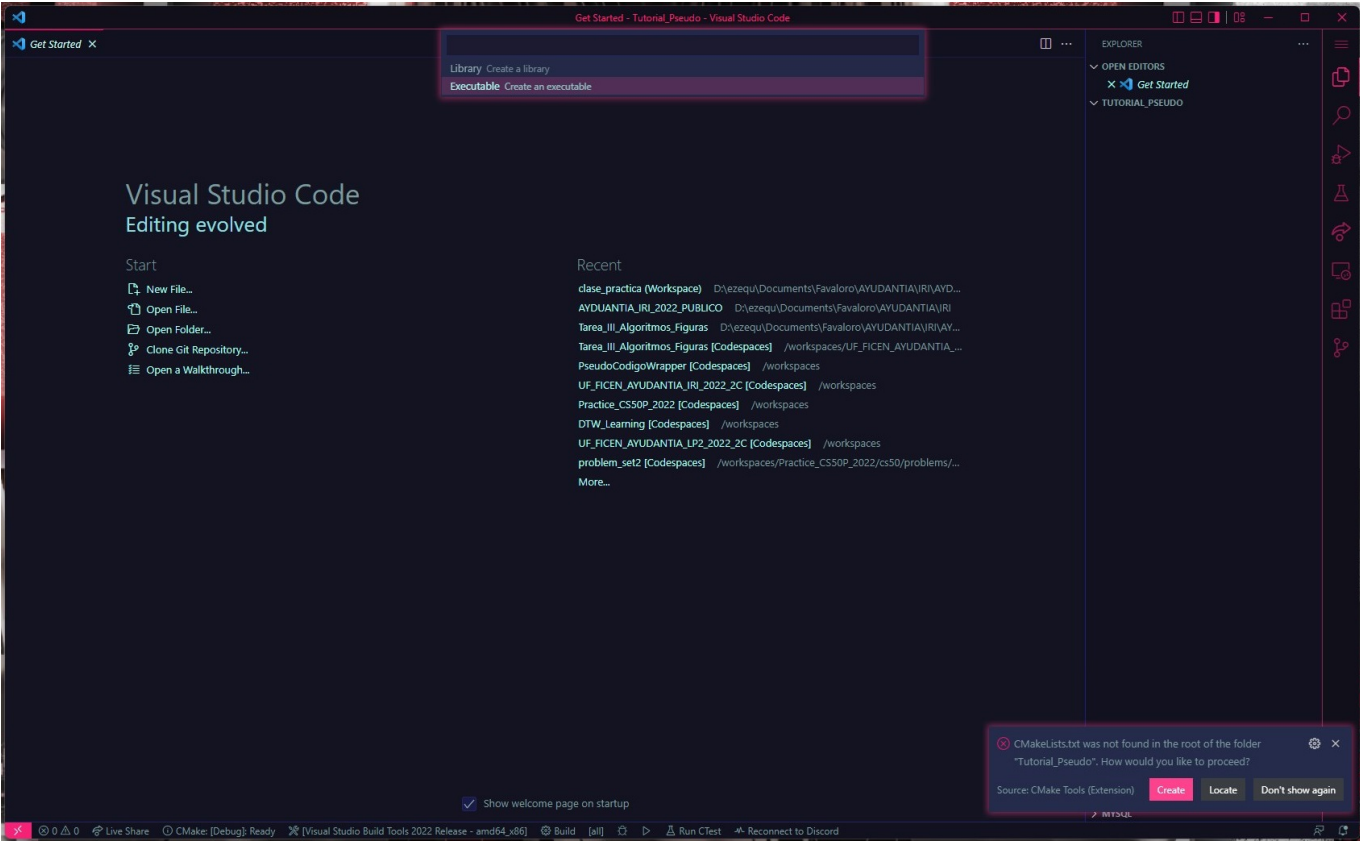
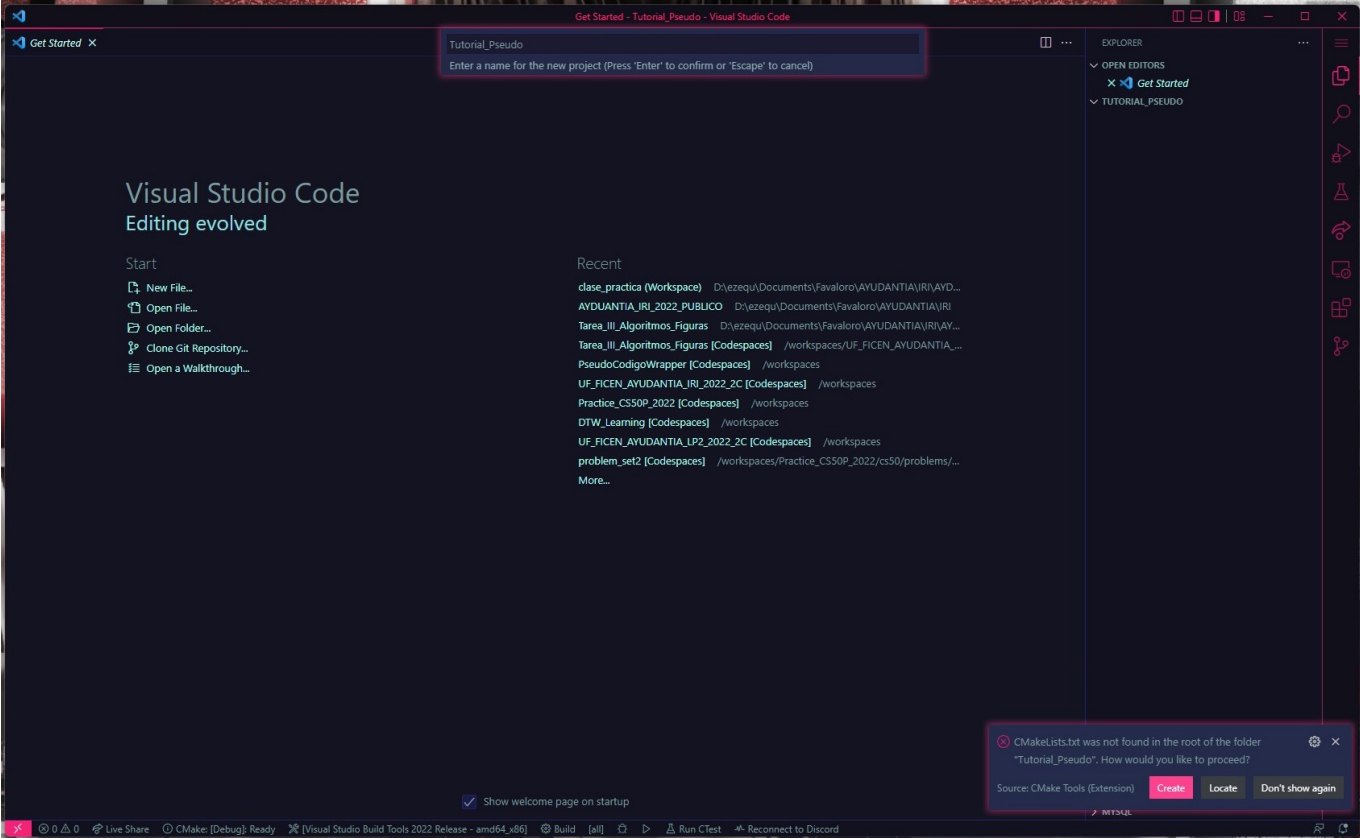


Visual Studio Code









The screenshot shows the Visual Studio Code interface. The main editor displays a file named `CMakeLists.txt` with the following content:

```
1 cmake_minimum_required(VERSION 3.0.0)
2 project(Tutorial_Pseudo VERSION 0.1.0)
3
4 include(CTest)
5 enable_testing()
6
7 add_executable(Tutorial_Pseudo main.cpp)
8
9 set(CPACK_PROJECT_NAME ${PROJECT_NAME})
10 set(CPACK_PROJECT_VERSION ${PROJECT_VERSION})
11 include(CPack)
12
```

The right sidebar shows the Explorer view with the project structure:

- OPEN EDITORS
 - `CMakeLists.txt`
- TUTORIAL_PSEUDO
 - build
 - `CMakeLists.txt`
 - `main.cpp`

The bottom panel shows the Output view with the CMake build log:

```
[proc] Executing command: "C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\Common7\IDE\CommonExtensions\Microsoft\CMake\bin\cmake.exe"
--no-warn-unused-cli -DCKMAKE_EXPORT_COMPILE_COMMANDS=BOOL=TRUE -Sd:/ezeug/Documents/Favaloro/AYUDANTIA/Tutorial_Pseudo -Bd:/ezeug/Documents/Favaloro/AYUDANTIA/
Tutorial_Pseudo/build -G "Visual Studio 17 2022"
[cmake] Not searching for unused variables given on the command line.
[cmake] -- The C compiler identification is MSVC 19.33.31629.0
[cmake] -- The CXX compiler identification is MSVC 19.33.31629.0
[cmake] -- Detecting C compiler ABI info
[cmake] -- Detecting C compiler ABI info - done
[cmake] -- Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2022/BuildTools/VC/Tools/MSVC/14.33.31629/bin/Hostx64/x64/cl.exe - skipped
[cmake] -- Detecting C compile features
[cmake] -- Detecting C compile features - done
[cmake] -- Detecting CXX compiler ABI info
[cmake] -- Detecting CXX compiler ABI info - done
[cmake] -- Check for working CXX compiler: C:/Program Files (x86)/Microsoft Visual Studio/2022/BuildTools/VC/Tools/MSVC/14.33.31629/bin/Hostx64/x64/cl.exe - skipped
[cmake] -- Detecting CXX compile features
[cmake] -- Detecting CXX compile features - done
[cmake] -- Configuring done
[cmake] -- Generating done
[cmake] -- Build files have been written to: D:/ezeug/Documents/Favaloro/AYUDANTIA/Tutorial_Pseudo/build
[ctest] There was an error running ctest to determine available test executables
```

Bajar la libreria

- Para bajar la libreria, haga click [AQUI](#)
- Descargue el archivo [Source code \(zip\)](#)
- Descomprima el archivo dónde usted quiera, pero que recuerde como volver.
 - Si no sabe como descomprimir un archivo, aquí le dejo un tutorial [Archivos ZIP: como Extraer o Descomprimir archivos en mi pc Windows 10 2022 sin programas de Erick Clavijo Morante](#)
- El archivo descomprimido deberia tener este formato

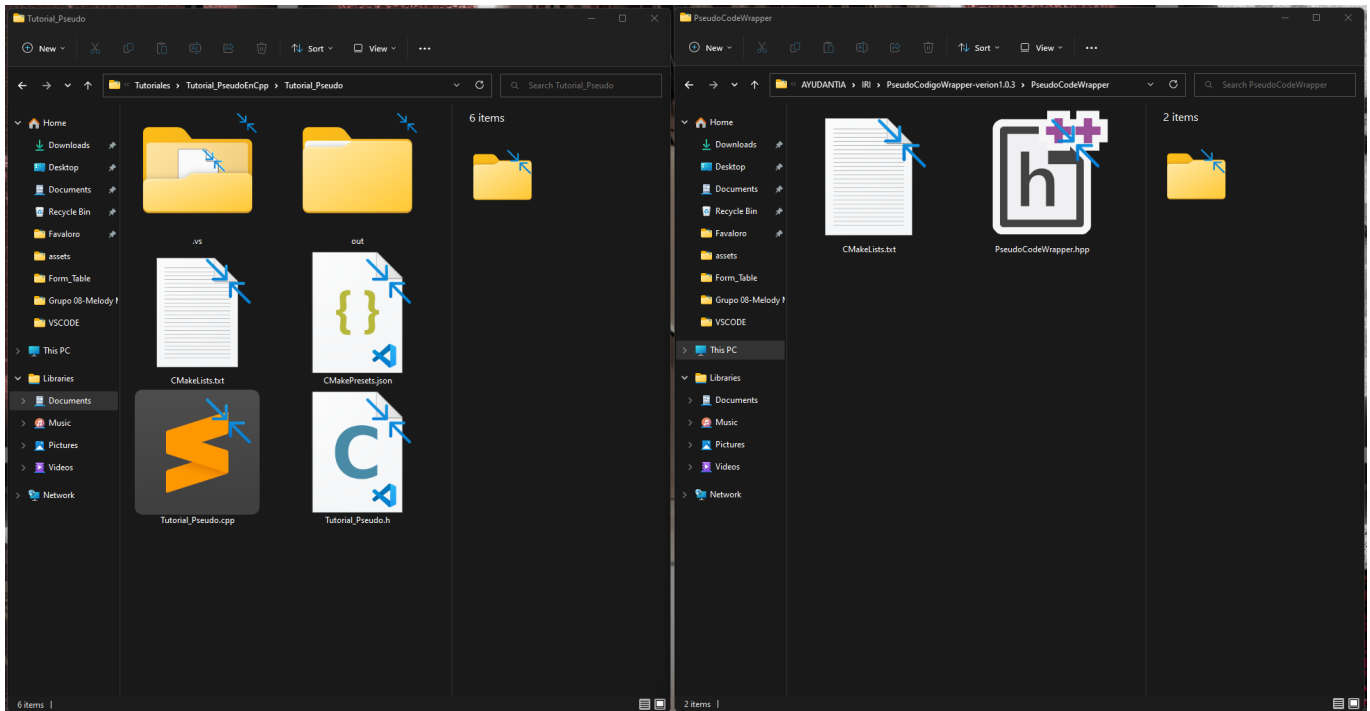
```
PseudoCodigoWrapper-verion1.0.3
+-- PseudoCode
|   +-- CMakeLists.txt
|   +-- PseudoCodeWrapper.hpp
+-- demo
|   +-- <Archivos varios>
+-- test
|   +-- <Archivos varios>
+-- <Archivos varios>
```

- Acceda a la carpeta ya descomprimida y entre a PseudoCodeWrapper
 - Dentro de está carpeta tendrá que haber dos archivos:
 - [CMakeLists.txt](#)
 - [PseudoCodeWrapper.hpp](#)

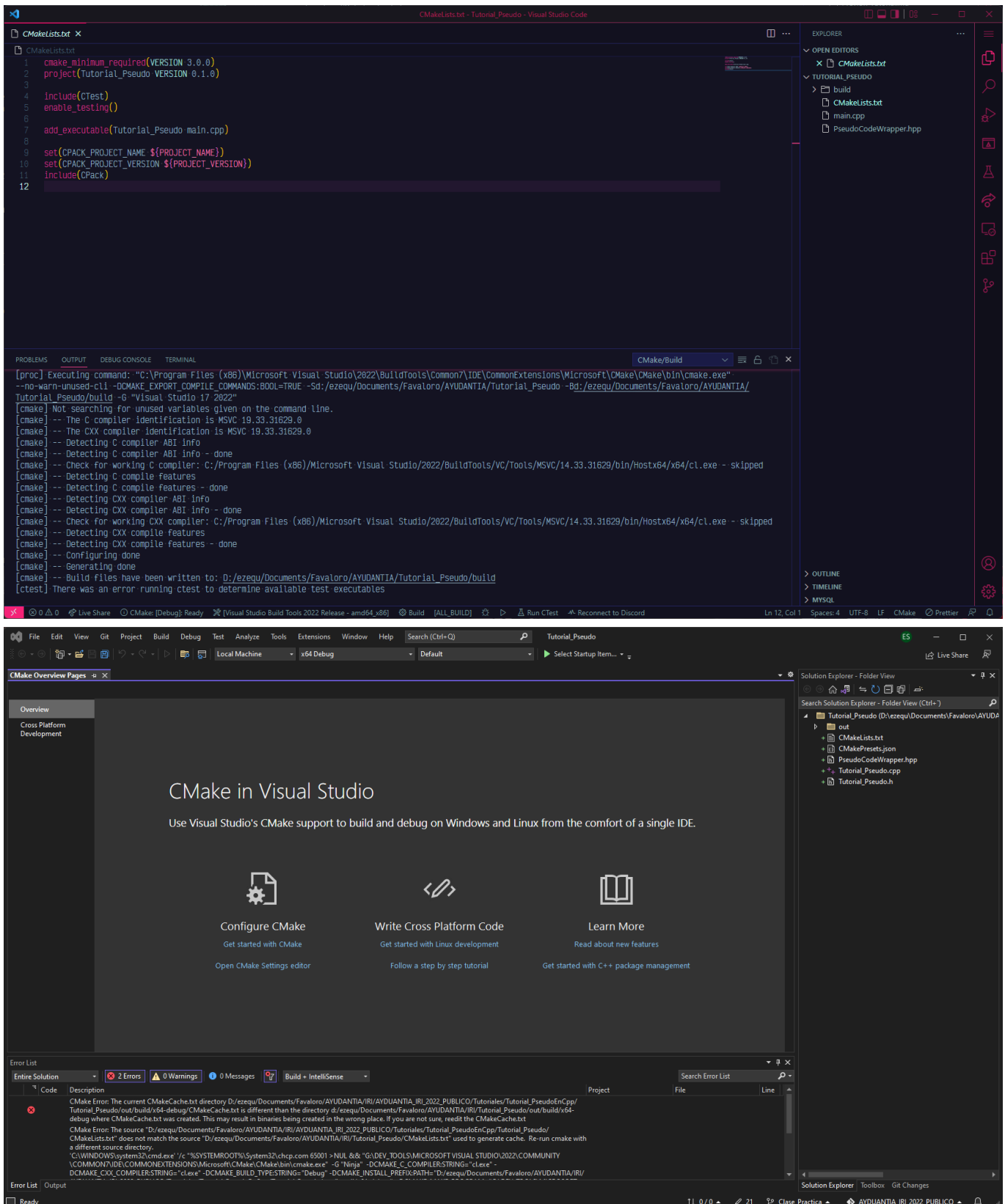
Instalar libreria dentro del proyecto

Este proceso aplica igual para Visual Studio 2022 y Visual Studio Code

- Abrimos la carpeta dónde se encuentra el proyecto y abrimos también la carpeta dónde está guardado el archivo **PseudoCodeWrapper.hpp** (esto ya lo hicimos unos momentos antes).



- Copiamos el archivo **PseudoCodeWrapper.hpp**, dentro de la carpeta del proyecto que hicimos. Si este paso se realizó de forma correcta, deberíamos verlo en el explorador de soluciones del Visual Studio Code como en el Visual Studio 2022
⚠ Si el Visual Studio 2022 tirar error, no nos alarmemos, ya lo solucionaremos en el siguiente paso!



- Ahora viene el paso más importante.
Abrimos el archivo **CMakeLists.txt** y añadimos el nombre **PseudoCodeWrapper.hpp** en `add_executable()`

Este CMakeLists.txt puede variar al que tengas, pero el formato debería ser similar

```
# CMakeList.txt : CMake project for Tutorial_Pseudo, include source and define
# project specific logic here.
```

```
#
cmake_minimum_required (VERSION 3.8)

project ("Tutorial_Pseudo")

# Add source to this project's executable.
add_executable (Tutorial_Pseudo "Tutorial_Pseudo.cpp" "Tutorial_Pseudo.h"
"PseudoCodeWrapper.hpp")

if (CMAKE_VERSION VERSION_GREATER 3.12)
    set_property(TARGET Tutorial_Pseudo PROPERTY CXX_STANDARD 20)
endif()

# TODO: Add tests and install targets if needed.
```

- Ahora hacemos `#include "PseudoCodeWrapper.hpp"` en nuestro `main.cpp` o dónde tengamos la función `int main()`
- Felicidades, añadiste de forma exitosa la librería!!!

En un futuro cercano, tendremos más cercanía con CMake y sus utilidades.