

Configuración básica del entorno de desarrollo

En el amplio campo de la programación competitiva, C++ es uno de los lenguajes más utilizados por la velocidad que ofrece frente a otros lenguajes, así como las funciones integradas en su librería estándar (STL).

Por este motivo, procederemos a instalar un compilador de C++ para tener un acceso a toda la librería estándar, la cual nos facilitará realizar varios ejercicios de programación competitiva.

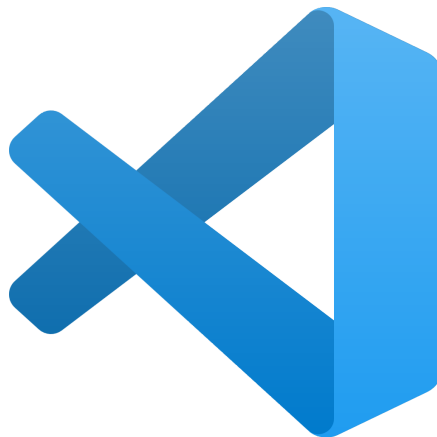
Para esto, necesitaremos instalar los siguientes dos softwares:

- **Compilador:** GCC (G++) en su versión más actualizada.
- **Editor de texto:** Visual Studio Code

Nota:

Recordemos que **Visual Studio** **NO** es lo mismo que **Visual Studio Code**. El **primero** es un IDE (Entorno de desarrollo integrado) y el **segundo** un editor de texto.

Para nuestro caso, utilizaremos **Visual Studio Code** por ser más ligero, veloz y adecuado que **Visual Studio** para resolver ejercicios de programación competitiva.



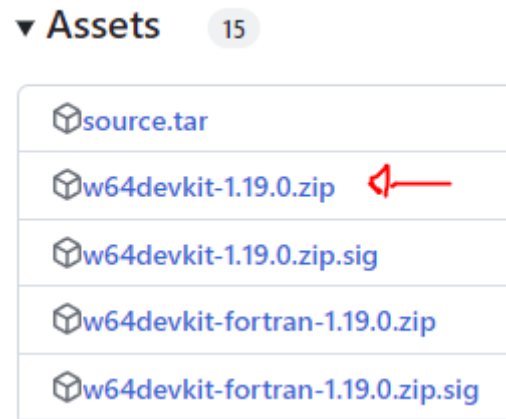
INSTALACIÓN DEL COMPILADOR

Accedemos al repositorio de Github: <https://github.com/skeeto/w64devkit>

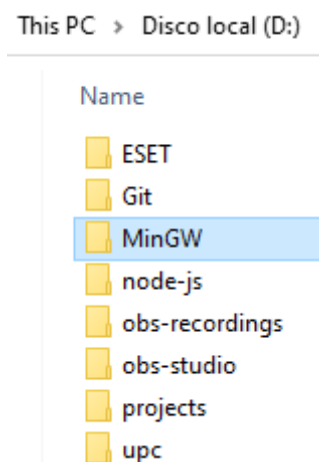
1. En la parte lateral derecha, daremos click en el **release** con la versión más actualizada que salga. En este caso, es **1.19.0**.



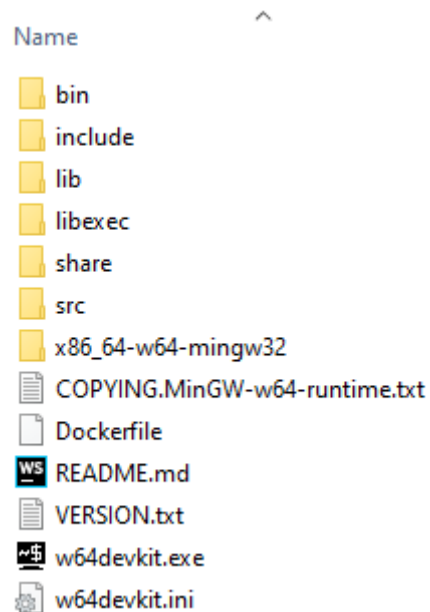
2. En la nueva vista que tengamos, seleccionaremos el segundo archivo **w64devkit-1.19.0.zip**:



3. Una vez descargado, **descomprimos** el archivo en una ruta conozcamos y no vayamos a modificar. Adicionalmente, podemos cambiar el nombre de la carpeta a **MinGW**.

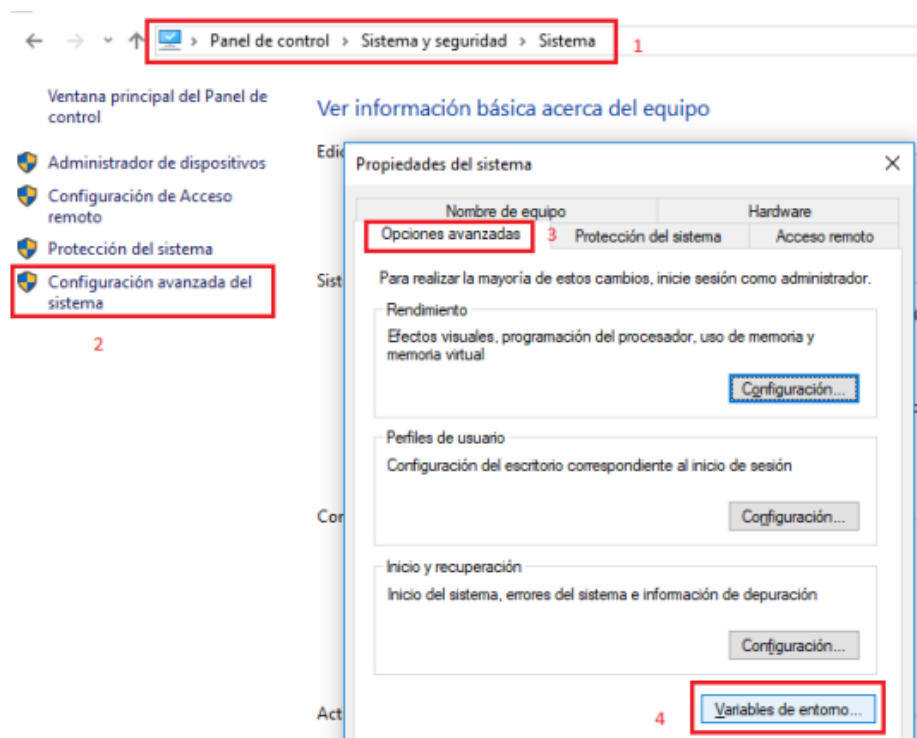


4. Dentro de esa carpeta, deberemos tener una relación similar a la siguiente:

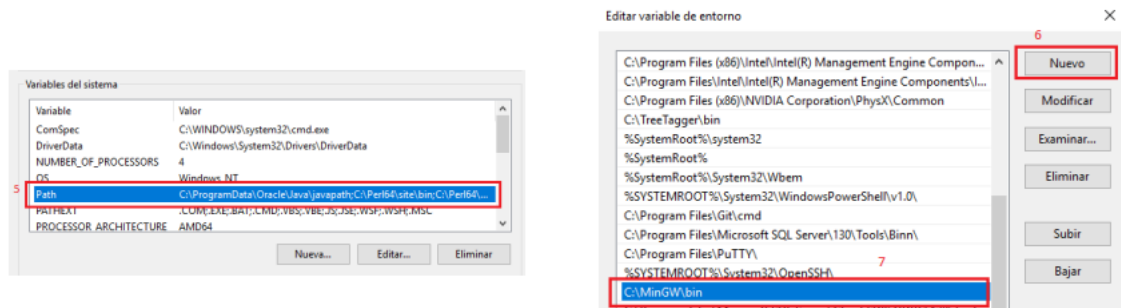


5. Seguidamente, agregamos la dirección/path de la carpeta **bin** a la variables de entorno. Para eso, tendremos que ir a:

Panel de Control > Sistema y Seguridad > Sistema > Configuración Avanzada del sistema > Opciones avanzadas > Variables de entorno



6. Aquí (en variables del sistema), daremos doble click a **PATH** y agregaremos la ruta de la carpeta **bin** del **MinGW**.



7. Finalmente, si abrimos una terminal y probamos el comando de **g++** deberíamos tener un mensaje similar al siguiente:

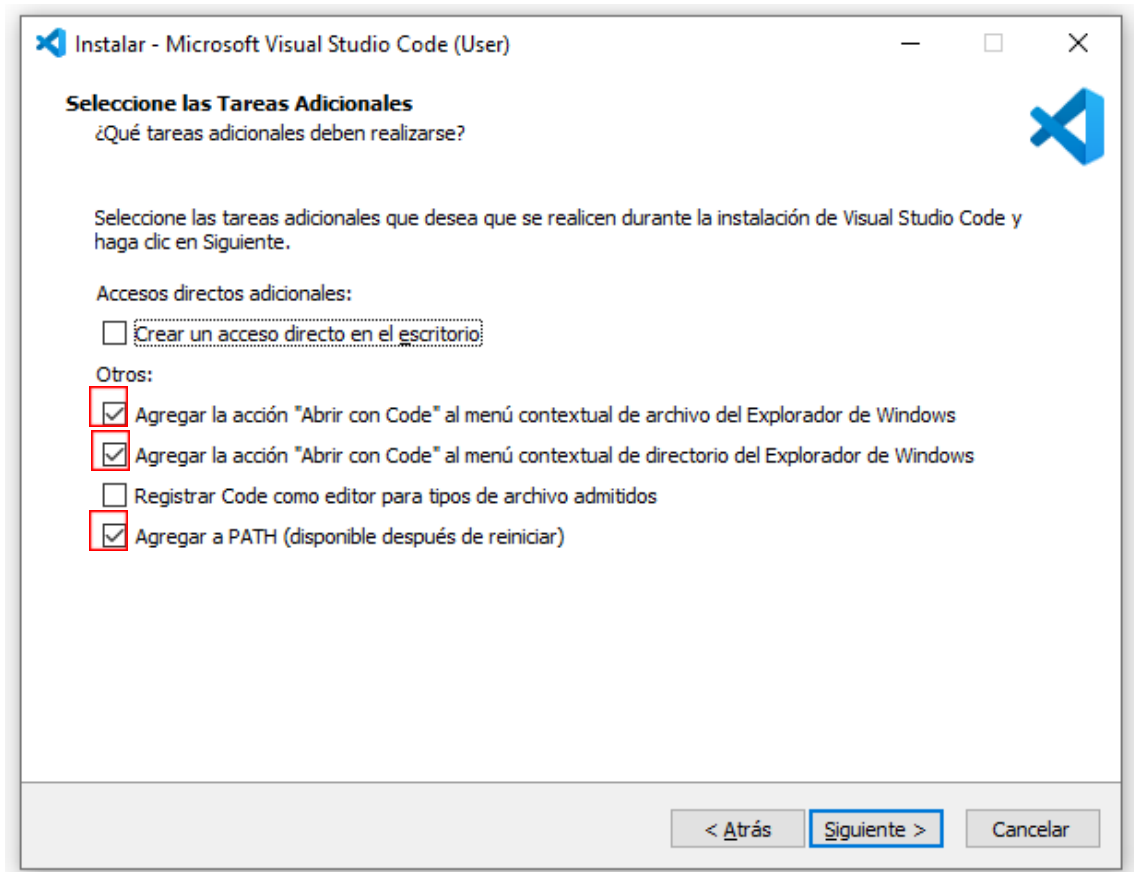
```
C:\Users\Dobby>g++  
g++: fatal error: no input files  
compilation terminated.
```

Con esto, ya estaríamos listos para empezar a configurar el editor de texto para que nos dé soporte al trabajar con C/C++.

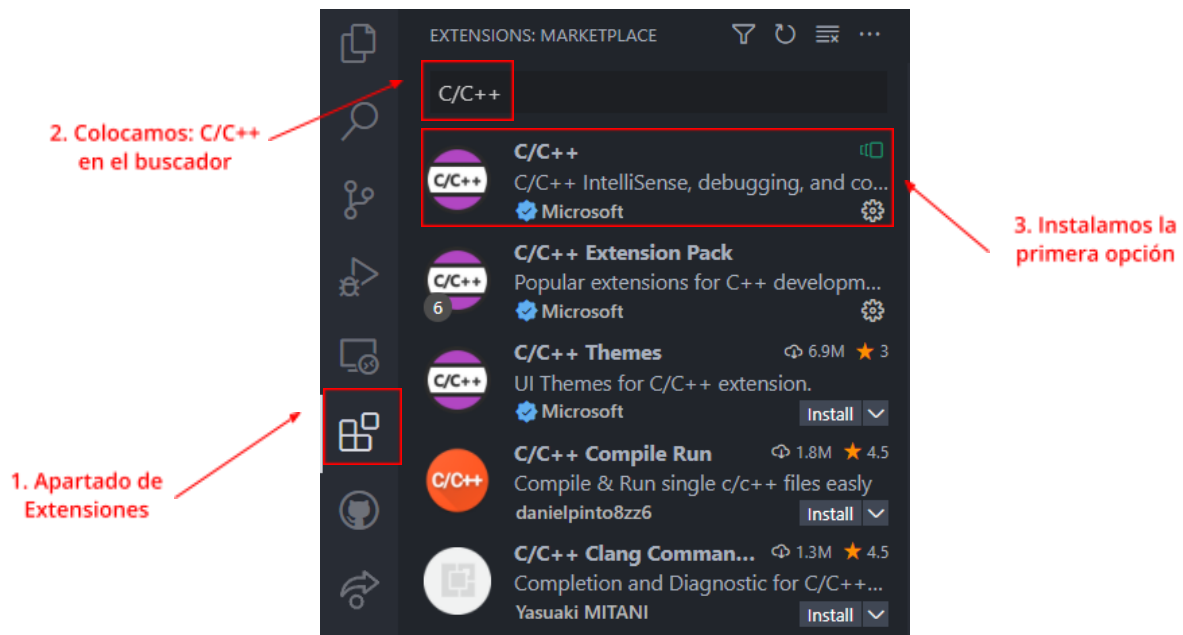


INSTALACIÓN Y CONFIGURACIÓN DEL EDITOR DE TEXTO

1. Descargamos [Visual Studio Code](https://code.visualstudio.com/) desde: <https://code.visualstudio.com/> y lo instalamos de manera normal asegurándonos que estén seleccionadas las siguientes tres opciones.

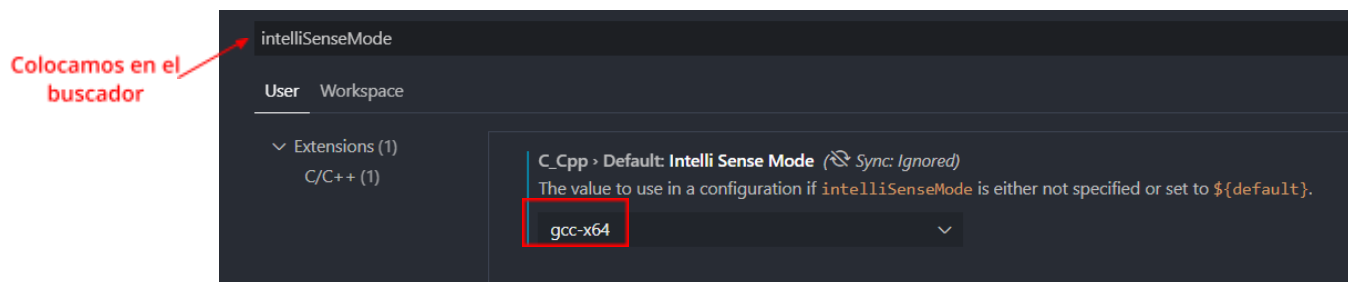


2. Abrimos el editor de texto e instalamos la extensión de Microsoft para C/C++ ya que nos brindará un IntelliSense.



3. Después de instalar la extensión, colocaremos el parámetro de configuración **intelliSenseMode** en la opción de **gcc-x64**.

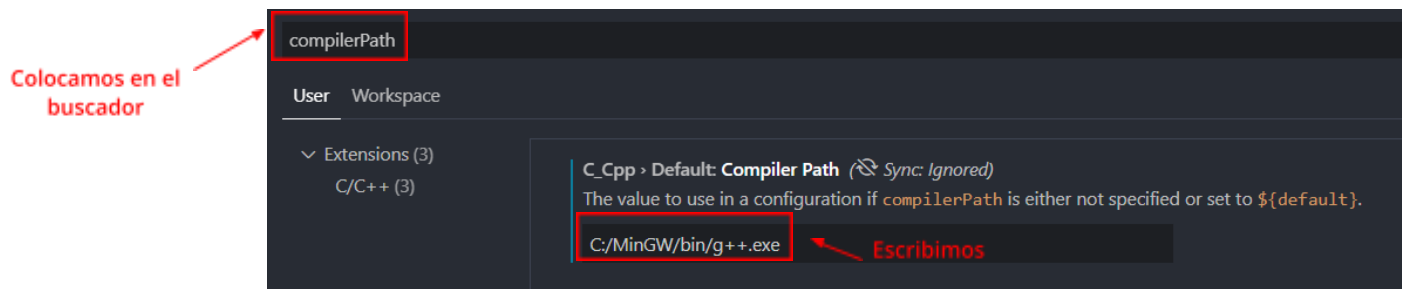
Vamos a File > Preferences > Settings y colocamos en el buscador: **intelliSenseMode**.



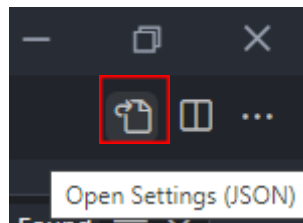
4. Procederemos a cambiar el PATH de compilación de VSCode. Para eso, colocaremos la ruta del compilador G++ que instalamos.

Vamos a File > Preferences > Settings

Y escribimos lo siguiente al buscar **compilerPath** en el buscador:



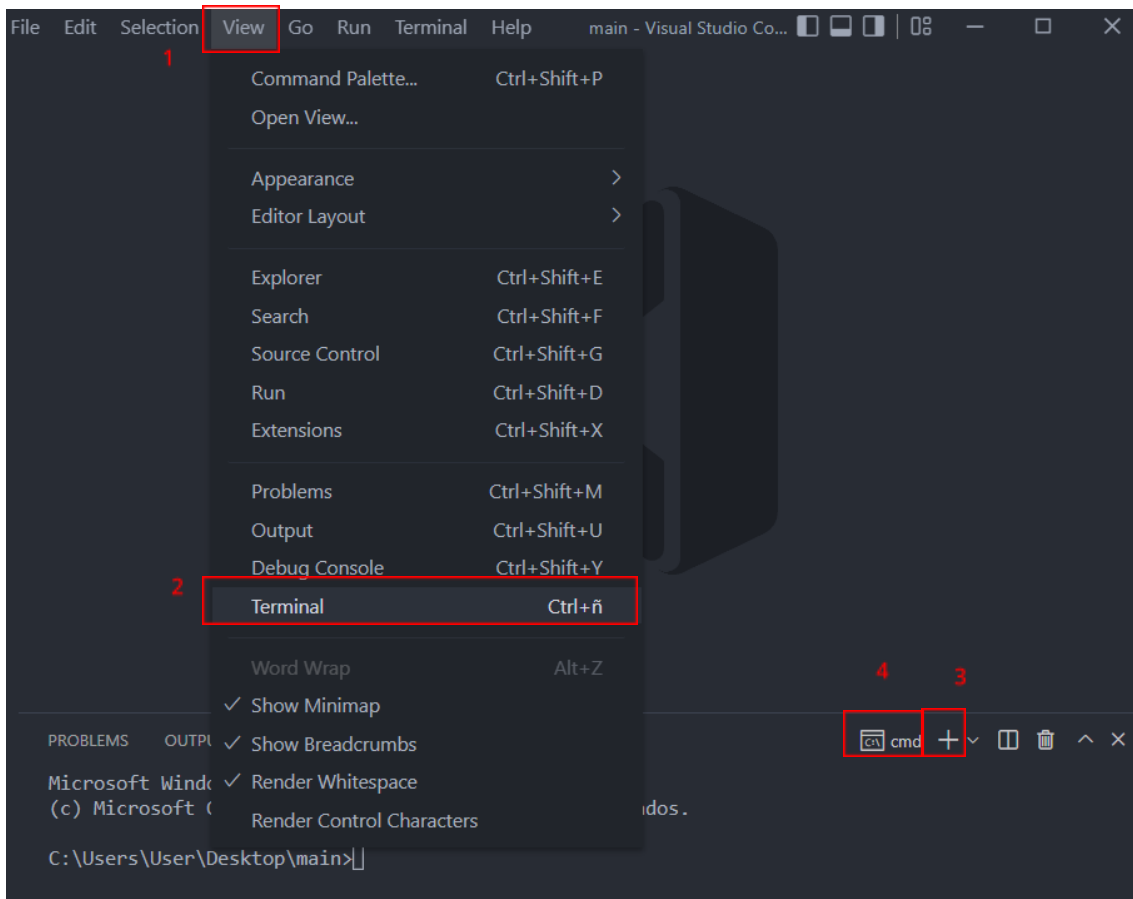
Si por algún motivo no pudiésemos escribir en el recuadro, iremos al extremo derecho de la ventana y daremos click en **Open Settings JSON**



Y buscaremos la línea que comience con **C_Cpp.default.compilerPath** y escribiremos la ruta:

```
"C_Cpp.default.compilerPath": "C:/MinGW/bin/g++.exe",
```

5. Finalmente, agregaremos una terminal de VSCode que sea del tipo **cmd** para poder realizar la compilación y ejecución de nuestro programa por ese medio.



Llegados a este punto, realizaremos una prueba para ver el flujo de trabajo, así como conocer como se realiza el proceso de compilación y ejecución cuando trabajemos con C/C++ en Visual Studio Code.

PROCESO DE COMPILACIÓN

El proceso de compilación se basa en que el compilador recibe un **código fuente** (en un lenguaje de alto nivel) para que este sea traducido a un **archivo ejecutable** (en lenguaje máquina).



PROCESO DE EJECUCIÓN

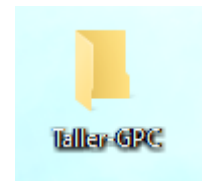
Una vez que obtenemos nuestro archivo ejecutable, natural y redundantemente, lo ejecutaremos para utilizarlo con unos **datos de entrada** y que nos dé **datos de salida**.



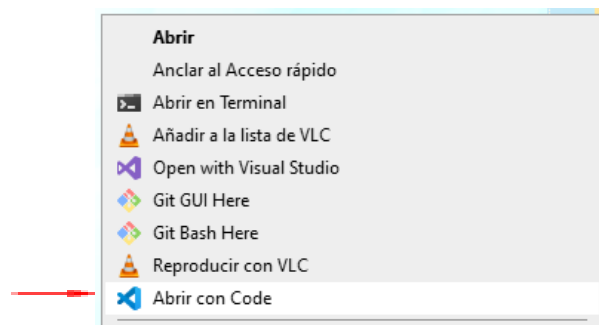
FLUJO DE TRABAJO PROBANDO LA COMPILACIÓN Y EJECUCIÓN

A continuación, mostraremos el flujo de trabajo para probar ambos procesos.

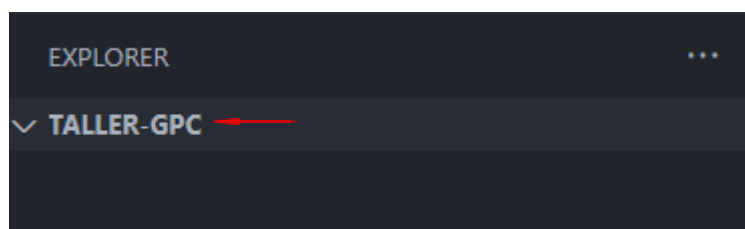
Empezaremos por crear una carpeta donde vamos a guardar los archivos .cpp que vayamos escribiendo. En este caso, se creó una carpeta llamada **"Taller-GPC"** en el escritorio.



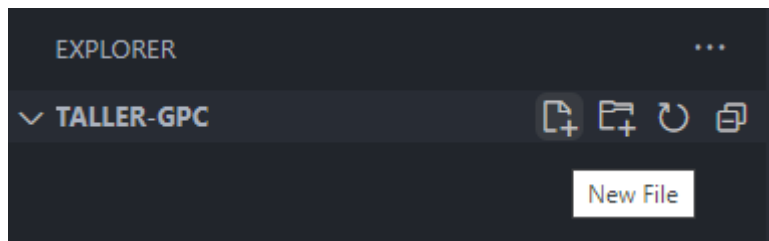
Si le damos click derecho a la carpeta, veremos que nos sale una opción **"Abrir con Code"**. Procederemos a darle click a esa opción.



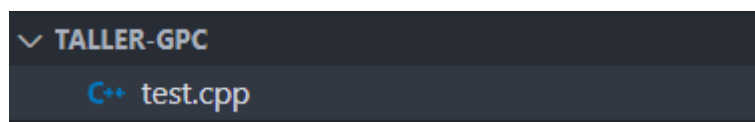
Inmediatamente, se abrirá Visual Studio Code ubicado en esa carpeta. Sabremos que estamos dentro de ella porque en el explorador de archivos de la izquierda saldrá el nombre de la carpeta:



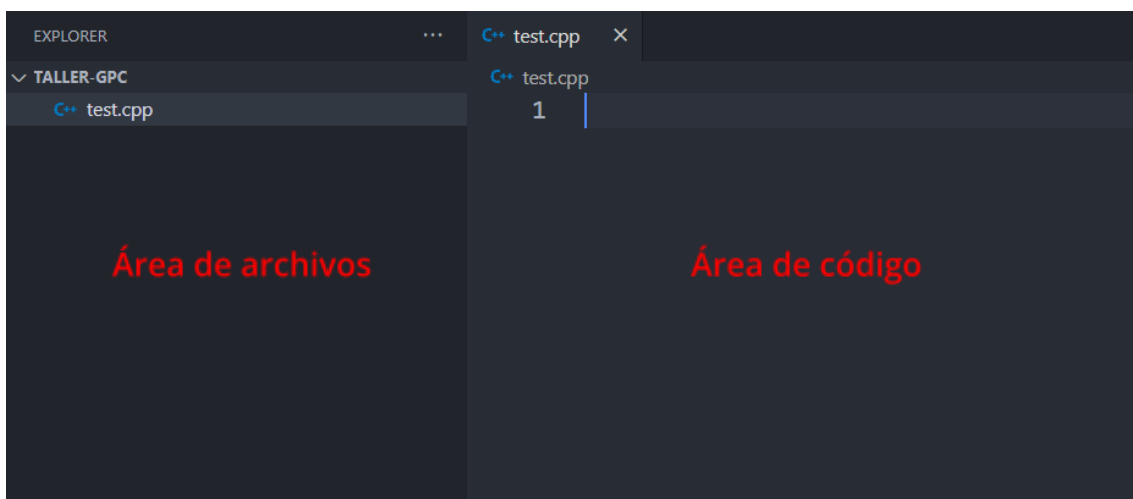
Para agregar un archivo, nos ubicaremos en el explorador de archivos (de VSCode), y, con el cursor encima del nombre de la carpeta, se nos habilitará varias opciones donde seleccionaremos **New File**:



Se nos creará un archivo al cual nosotros le podremos dar **un nombre y la extensión del archivo**. En este caso, lo nombraremos **test.cpp**



Como resultado, veremos que a la derecha se nos abre el archivo para que podamos escribir sobre él. Aquí irá nuestro código.



A continuación, escribiremos la aplicación sencilla de **hello world**.

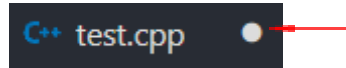
```
C++ test.cpp x
C++ test.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      cout << "Hello GPC!";
7      return 0;
8  }
```

Nos aseguramos de guardar el archivo una vez hallamos terminado de escribir en él.

Para eso, la forma más rápida es utilizar el atajo de teclado:

Ctrl + S

Nota: Sabremos que debemos guardar el archivo porque saldrá un círculo de color blanco en el nombre del archivo, en vez de la típica x.



Para compilar nuestro código y obtener un ejecutable, realizaremos lo siguiente:

1. Abrimos una terminal en VSCode (Ctrl + ñ para Español, Ctrl + ~ para Inglés).
2. Escribimos: `g++ nombre_del_archivo.cpp`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [Versión 10.0.19044.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

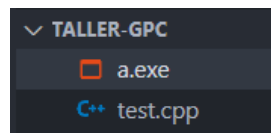
C:\Users\User\Desktop\Taller-GPC>g++ test.cpp
```

3. Damos enter y esperamos que termine la compilación. Sabremos que ha terminado porque el cursor estará listo para escribir otra instrucción en la terminal.

```
C:\Users\User\Desktop\Taller-GPC>g++ test.cpp

C:\Users\User\Desktop\Taller-GPC>
```

Además de poder ver el archivo ejecutable creado: **a.exe** en el explorador de archivos de la izquierda.



Para ejecutar nuestro código, en la misma terminal que tenemos escribimos: **a.exe**

```
Microsoft Windows [Versión 10.0.19044.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\User\Desktop\Taller-GPC>g++ test.cpp

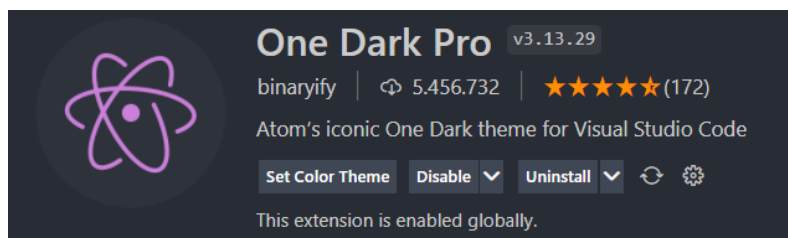
C:\Users\User\Desktop\Taller-GPC>a.exe
```

Una vez demos enter, se ejecutará y se mostrará la salida datos en la consola.

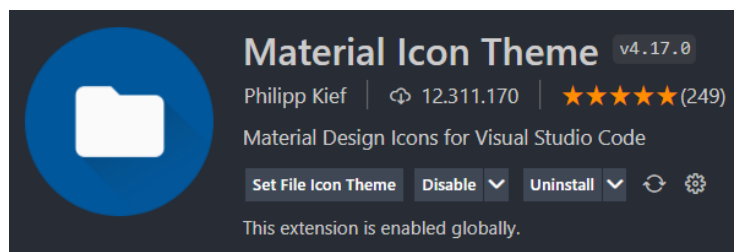
```
C:\Users\User\Desktop\Taller-GPC>a.exe
Hello GPC!
```

(OPCIONAL) PERSONALIZANDO VISUAL STUDIO CODE

Si queremos cambiar el tema que viene por defecto, podemos buscar en la tienda de extensiones e instalar: **One Dark Pro**. Una vez se instale, seleccionar la primera opción.



Si queremos tener mejores íconos en nuestro explorador de archivos, podemos buscar en la tienda de extensiones e instalar: **Material Icon Theme**.



ATAJOS DE TECLADOS BÁSICOS Y ÚTILES EN VSCODE

Ctrl + B	Abrir o Cerrar explorador de archivos.
Ctrl + S	Guardar archivo.
Ctrl + Ñ	Abrir o Cerrar la terminal.
Ctrl + X	Cortar línea de código.
Alt + Abajo	Mover línea de código hacia abajo.
Alt + Arriba	Mover línea de código hacia arriba.
Alt + Shift + Abajo	Duplicar línea de código y bajar el cursor.
Alt + Shift + Arriba	Duplicar línea de código sin bajar el cursor.

COMANDOS BÁSICOS EN CONSOLA

cls	Limpiar la consola
Flecha Arriba	Volver al comando anteriormente escrito
Flecha Abajo	Volver al comando posteriormente escrito
Tab	Permite autocompletar comando en la consola

CONTENIDO ELABORADO POR

- Bachelor y Coach del GPC UPC: Rodolfo Mercado Gonzales.
- Miembro Senior del GPC UPC: Andrés Leiva Bautista.