

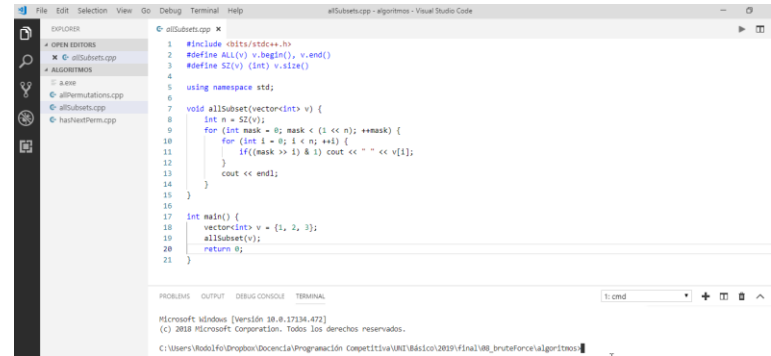
Ambiente de trabajo

❑ Editor de Texto

Visual Studio Code

❑ Compilador GCC (g++)

- Windows: MinGW (versión GCC para Windows)

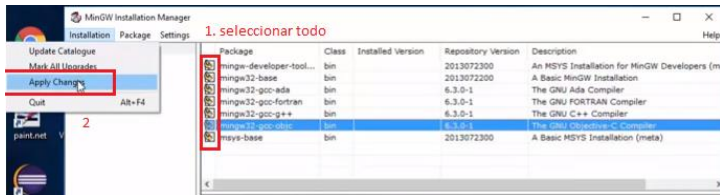


```
1 #include <iostream>
2 #define ALL(v) v.begin(), v.end()
3 #define SZ(v) (int) v.size()
4
5 using namespace std;
6
7 void allSubsets(vector<int> v) {
8     int n = SZ(v);
9     for (int mask = 0; mask < (1 << n); ++mask) {
10         for (int i = 0; i < n; ++i) {
11             if ((mask >> i) & 1) cout << " " << v[i];
12         }
13         cout << endl;
14     }
15 }
16
17 int main() {
18     vector<int> v = {1, 2, 3};
19     allSubsets(v);
20     return 0;
21 }
```

Instalación y Configuración del MinGW

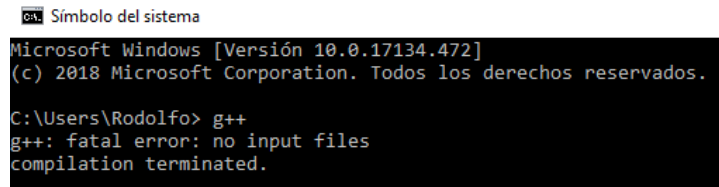
1. Instalar MinGW con las opciones por defecto.

Cuando nos consulte por los paquetes que queremos instalar, seleccionamos todos.

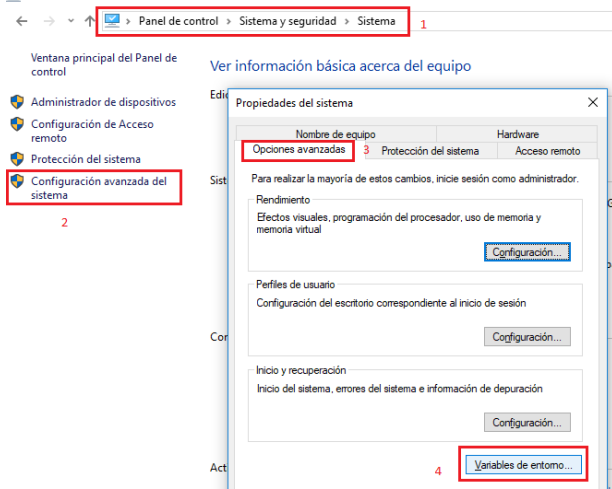


2. Verificar que el compilador esté en el *Path* del sistema:

Abrimos la terminal de Windows (window + R, escribimos cmd y damos enter), escribimos g++ y damos enter. Si nos aparece un mensaje como en la imagen entonces esta todo ok.



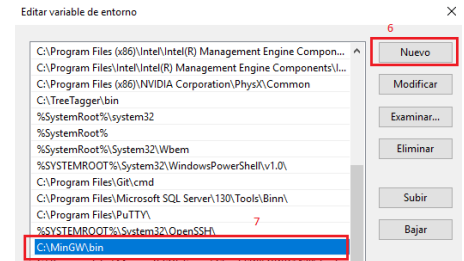
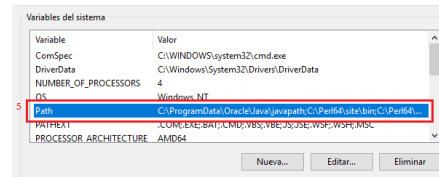
Instalación y Configuración del MinGW



3. En caso no sea exitoso el paso 2, tenemos que ir a :

Panel de control → Sistema y Seguridad → Sistema → Configuración avanzada del sistema → Opciones avanzadas → Variables de entorno.

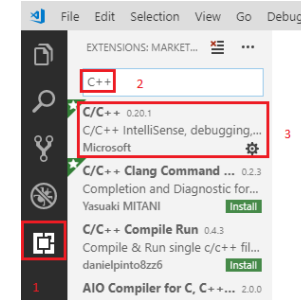
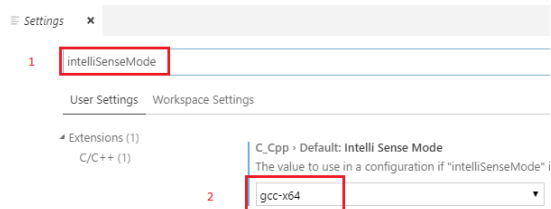
En variables del sistema damos doble click en *Path* y agregamos la ruta de la carpeta bin del MinGW.



Instalación y Configuración del VSCode

1. Instalar Visual Studio Code con las opciones por defecto.
2. Instalar la extensión de Microsoft para C/C++ (nos brinda IntelliSense)
3. Colocar el parámetro de configuración *intelliSenseMode* en la opción “gcc-x64”:

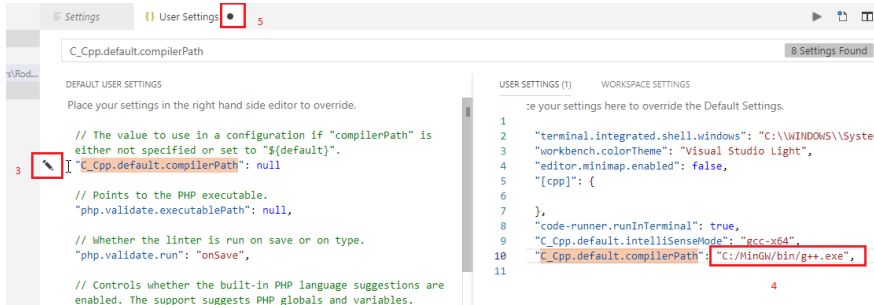
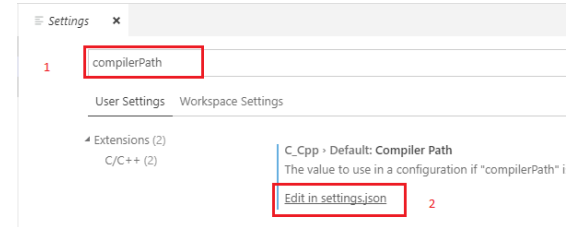
File → Preferences → Settings



Instalación y Configuración del VSCode

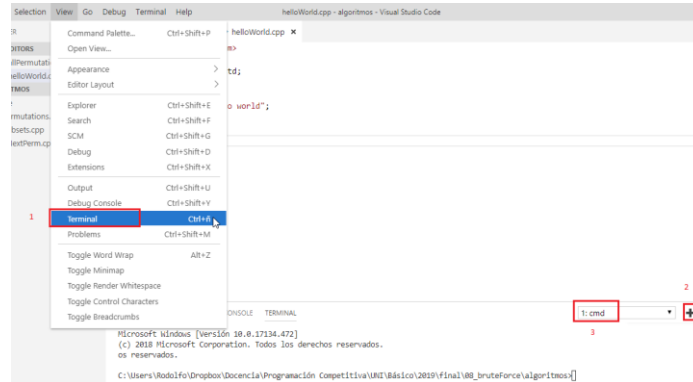
4. Cambiar el path de compilación que utiliza el VSCode, colocar la ruta del compilador g++ que instalamos.

File → Preferences → Settings

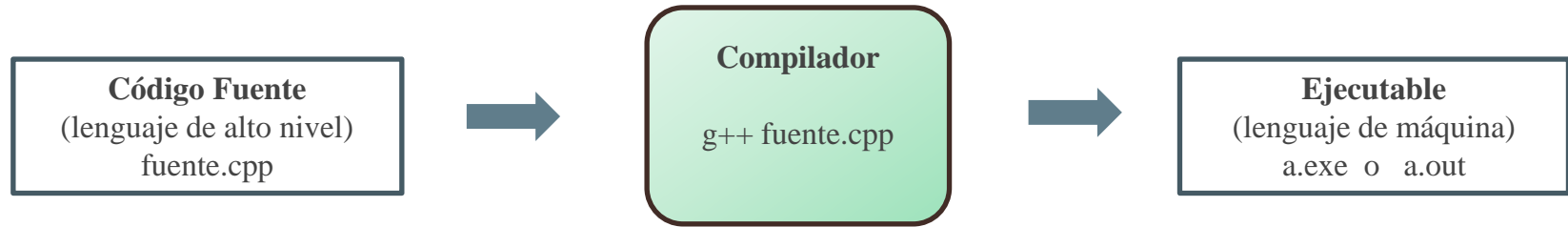


Configuración de la terminal

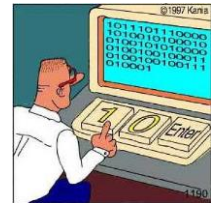
Agregamos una terminal de VSCode que sea del tipo “cmd” para poder realizar la compilación y ejecución de nuestro programa por ese medio .



Compilación

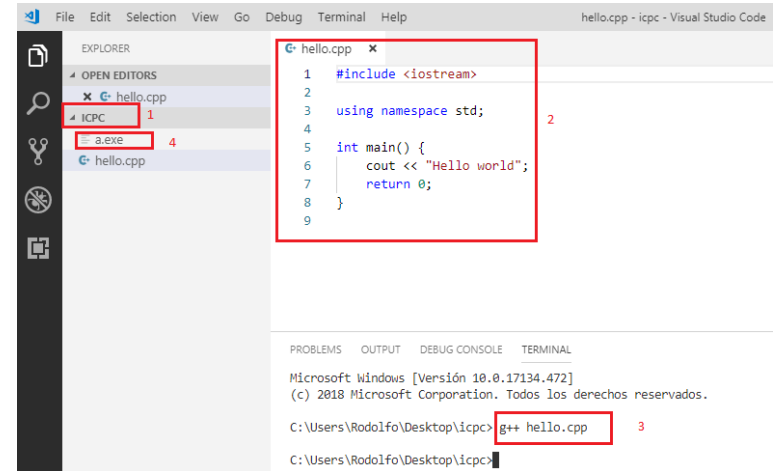


```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      cout << "Hello world";
7      return 0;
8  }
```

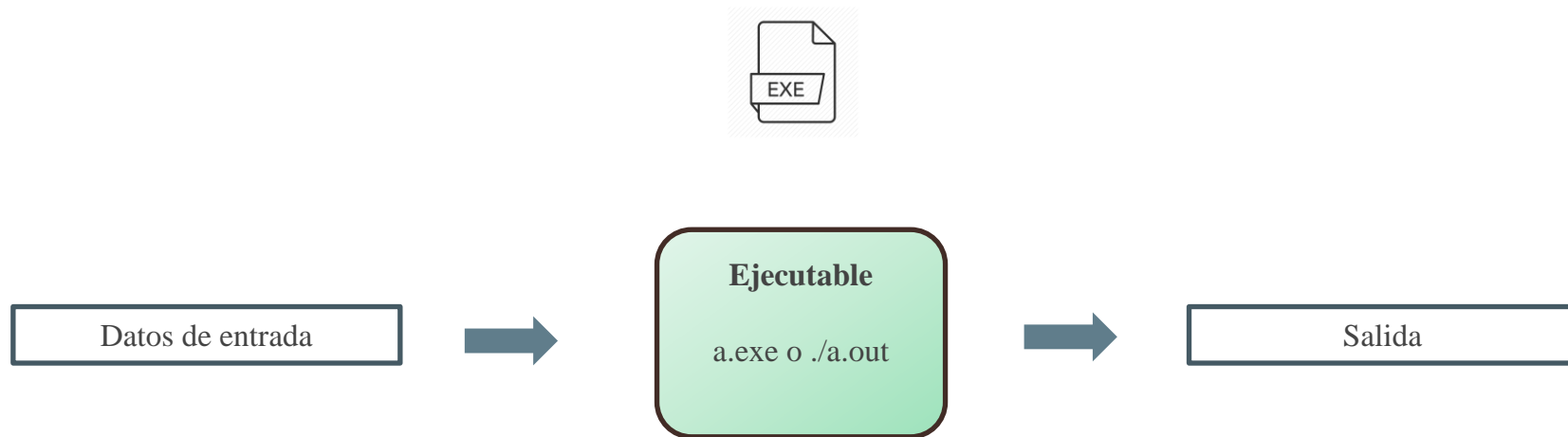


Compilación

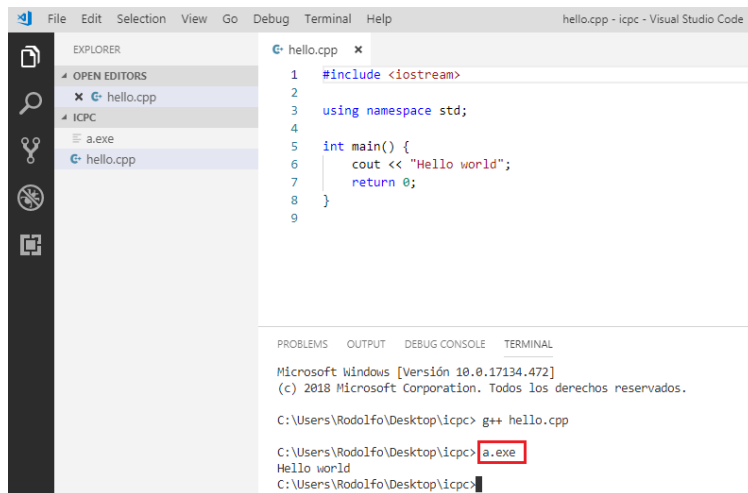
1. Es recomendable tener una carpeta de trabajo (en el ejemplo se llama ICPC) lo abrimos con File → Open Folder
2. Creamos un nuevo archivo en nuestra carpeta y lo guardamos con la extensión .cpp (en el ejemplo se llama hello.cpp)
3. Compilamos en la terminal.
4. Automáticamente se creó el ejecutable de nuestro programa.



Ejecución



Ejecución



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the file structure with 'hello.cpp' selected. The main editor window shows the source code of 'hello.cpp' with the following content:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     cout << "Hello world";
7     return 0;
8 }
9
```

Below the editor, the TERMINAL panel is active, showing the command prompt output:

```
Microsoft Windows [Versión 10.0.17134.472]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Rodolfo\Desktop\icpc> g++ hello.cpp

C:\Users\Rodolfo\Desktop\icpc> a.exe
Hello world
C:\Users\Rodolfo\Desktop\icpc>
```

The command 'a.exe' is highlighted with a red box in the terminal output.

