



En este apunte trataremos el tema de instalación y configuración de algunas herramientas de desarrollo. No se pretende un texto completo y exhaustivo, pero si que sirva de guía, haciendo referencia a varios sitios en la web.

Versión 3 – fecha 15/04/2024

Visual Estudio Code

No es un IDE en el sentido tradicional, sino que sigue una tendencia actual que consiste en tener un muy buen editor con capacidad de recibir extensiones que aumenten sus capacidades. Entonces dependiendo del tipo de desarrollo que se quiera hacer hay que instalar las diferentes extensiones. Esto le da una gran adaptabilidad.

Cuesta un poco más configurarlo, pero es muy versátil, es multiplataforma (windows, linux, mac) y tiene buena integración con git (en particular con github). Es la herramienta recomendada para nuestro curso.

Sitio

<https://code.visualstudio.com/>

Desde el link anterior da la posibilidad de bajar la versión correspondiente al sistema operativo que uses para visitar el sitio, pero en este link podés bajar cualquiera de las versiones.

<https://code.visualstudio.com/#alt-downloads>

La documentación de como instalar, configurar y usar está en

<https://code.visualstudio.com/docs>

El vínculo indicado arriba tiene toda la información, pero puede resultar muy laborioso seguir en orden, entonces, para simplificar, vamos a indicar los pasos y donde hay información para cada uno de ellos

Paso 1: Instalar el vscode. La documentación está en

Para windows

<https://code.visualstudio.com/docs/setup/windows>

Para linux

<https://code.visualstudio.com/docs/setup/linux>

Para mac

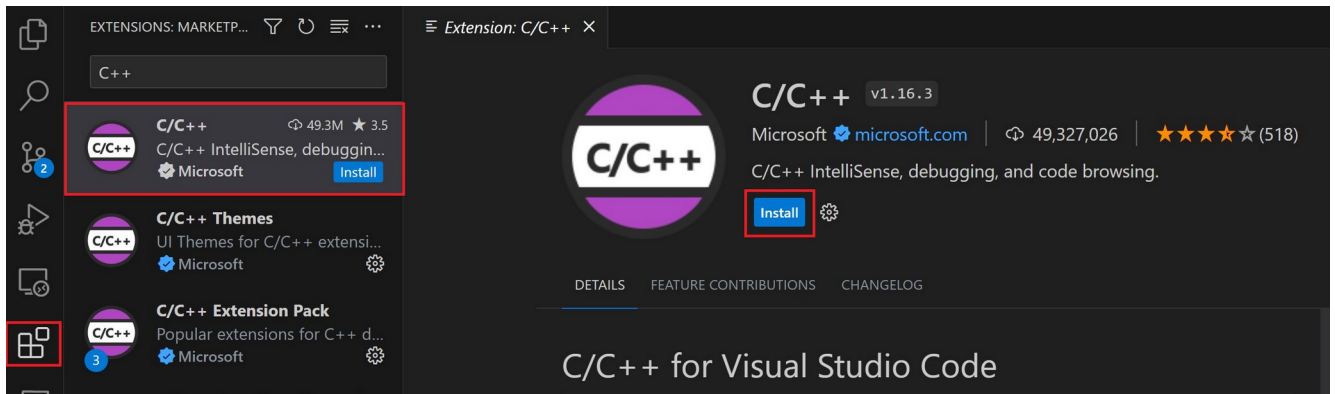
<https://code.visualstudio.com/docs/setup/mac>

Paso 2: instalar la extensión de C++

La documentación sobre como configurar el vscode de modo de poder desarrollar en C y C++ está en

<https://code.visualstudio.com/docs/languages/cpp>

Son varios pasos el primero es instalar la extensión propiamente dicha. Basta con instalar la extensión de C/C++, la que se ve como:



La que dice intellisense está obsoleta. Hay un pack recomendado que incluye la que mencionamos, C/C++ themes, que solo agrega un coloreado adicional a los fuentes y herramientas para manejar la herramienta de construcción CMAKE, que de todos modos no usaremos. El que quiera instalar cualquier paquete adicional, por supuesto que puede hacerlo, pero no es necesario.

Paso 3: El compilador

Depende de que sistema operativo uses, las instrucciones para tener el compilador y poder accederlo de visual code están en:

Para windows

<https://code.visualstudio.com/docs/cpp/config-mingw>

Para linux

<https://code.visualstudio.com/docs/cpp/config-linux>

Para mac

<https://code.visualstudio.com/docs/cpp/config-clang-mac>

En todos los casos lo importante es tener instalado el compilador, el debugger y la herramienta make. Si se tiene linux, siga las recomendaciones dadas por la documentación de su distribución, en general suele haber un meta paquete que instala todo o casi todo lo necesario, así según su distribución puede instalar desde consola :

Debian y derivados (el debugger gdb no es parte del meta paquete)

```
sudo apt install build-essential gdb
```

RedHat y derivados

```
sudo dnf install @development-tools
```

o bien

```
yum groupinstall "Development Tools"
```

OpenSuse

```
sudo zypper install -t pattern devel_basis
```

Arch y derivados

```
sudo pacman -Sy base-devel
```

Si tiene mac, debe instalar xcode y luego los compiladores, pero no tengo el conocimiento necesario para hacer alguna recomendación adicional.



Si tienen windows, la explicación que figura hoy día (abril de 2024) en la página de microsoft es la más conveniente (no siempre fue así). Hay muchas maneras de instalar el compilador, pero con msys2 tiene algunas ventajas adicionales.

Además de instalar el compilador hay que actualizar las variables de entorno como explica en el video que está en la página (por las dudas que lo saquen: <https://www.youtube.com/watch?v=oC69vIWofJQ>) Solo agregaría que conviene incluir dos path, el que indica el video "C:\msys64\ucrt64\bin" y también "C:\msys64\usr\bin". De este modo hay otras herramientas que utilizaremos más adelante que ya quedan habilitadas.

Resumen de los pasos a seguir y algunos detalles a tener en cuenta, si estás instalando en windows:

1. Primero instalar el MSYS2 <https://www.msys2.org/>
2. Instalar el compilador usando esta herramienta. Abra la consola MSYS2 UCRT64.
 1. Actualice con el comando: `pacman -Syuu`
 2. Instale con: `pacman -S --needed base-devel mingw-w64-x86_64-toolchain`



```
M ~
euzko@DESKTOP-48RF6P7 UCRT64 ~
$ pacman -S --needed base-devel mingw-w64-x86_64-toolchain
:: There are 19 members in group mingw-w64-x86_64-toolchain:
:: Repository mingw64
 1) mingw-w64-x86_64-binutils  2) mingw-w64-x86_64-crt-git  3) mingw-w64-x86_64-gcc
 4) mingw-w64-x86_64-gcc-ada  5) mingw-w64-x86_64-gcc-fortran
 6) mingw-w64-x86_64-gcc-libgfortran  7) mingw-w64-x86_64-gcc-libs  8) mingw-w64-x86_64-gcc-objc
 9) mingw-w64-x86_64-gdb  10) mingw-w64-x86_64-gdb-multiarch  11) mingw-w64-x86_64-headers-git
12) mingw-w64-x86_64-libgccjit  13) mingw-w64-x86_64-libmangle-git
14) mingw-w64-x86_64-libwinpthread-git  15) mingw-w64-x86_64-make  16) mingw-w64-x86_64-pkgconf
17) mingw-w64-x86_64-tools-git  18) mingw-w64-x86_64-winpthread-git
19) mingw-w64-x86_64-winstorecompat-git
Enter a selection (default=all):
resolving dependencies...
looking for conflicting packages...
```

3. Agregue el directorio donde instaló a la variable de entorno path, en ppio sería C:\msys64\ucrt64\bin (en algunos casos es necesario reiniciar el sistema, otras veces lo toma directo).
4. Conviene agregar también a path: C:\msys64\usr\bin

Una vez instalado todo, el vscode trabaja con base en un directorio (directorio de trabajo). Ahí tendrá el fuente a compilar. Hace falta crear los archivos .json en el directorio .vscode. Una vez hecho puede copiar ese directorio a otros para no tener que repetir estos pasos.

En el menú Terminal elegir el último ítem (configure default build task o su versión en castellano). Esto creará el archivo tasks.json, edítelo para agregar el standard a usar y que se requiere usar estrictamente el esándar.

Desde “Run and Debug” le ofrecerá crear un nuevo archivo launch.json, hágalo y luego copie los contenidos desde la página de microsoft (la que explica como instalar el compilador, cerca del final).

Puede que por diferencias de idioma no quede correctamente enganchada con la tarea de compilación, por ello debe verificar que el valor de “preLaunchTask” en launch.json coincida con el valor de “label” en tasks.json

Sobre como hacer debug

<https://code.visualstudio.com/docs/cpp/cpp-debug>

Sobre como usar sistemas de control de versiones

<https://code.visualstudio.com/docs/editor/versioncontrol>



Sobre dar formato al fuente

<https://code.visualstudio.com/docs/cpp/cpp-ide>

Formateador on-line (muy básico)

<https://codebeautify.org/c-formatter-beautifier>