

# UTILIZANDO O MICROSOFT VISUAL STUDIO CODE

Prof. Humberto Razente

Sala 1B144

# INSTALAÇÃO

- Instalar VS Code

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

- Instalar extensão para C/C++

- <https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools>

# INSTALAÇÃO

- Instalar MINGW-w64 com o MSYS2
  - [https://github.com/msys2/msys2-installer/releases/download/2022-01-18/msys2-x86\\_64-20220118.exe](https://github.com/msys2/msys2-installer/releases/download/2022-01-18/msys2-x86_64-20220118.exe)
- Abra um prompt de comandos e execute o comando do MSYS2
  - `cd c:\msys64\usr\bin`
  - `pacman -S --needed base-devel mingw-w64-x86_64-toolchain`



```
Command Prompt
C:\>cd c:\msys64\usr\bin
c:\msys64\usr\bin>pacman -S --needed base-devel mingw-w64-x86_64-toolchain_
```

# INSTALAÇÃO

- Tecle <ENTER> para instalar todos os pacotes

```
Command Prompt - pacman -S --needed base-devel mingw-w64-x86_64-toolchain

C:\>cd c:\msys64\usr\bin

c:\msys64\usr\bin>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): _
```

# INSTALAÇÃO

- Tecle <ENTER> novamente para instalar todos os pacotes

```
Command Prompt - pacman -S --needed base-devel mingw-w64-x86_64-toolchain
18) mingw-w64-x86_64-winpthreads-git 19) mingw-w64-x86_64-winstorecompat-git

Enter a selection (default=all):
resolving dependencies...
looking for conflicting packages...

Packages (64) binutils-2.37-5 bison-3.8.2-2 diffstat-1.64-1 diffutils-3.8-2
dos2unix-7.4.2-1 flex-2.6.4-2 gperf-3.1-3 groff-1.22.4-2 m4-1.4.19-2
make-4.3-3 mingw-w64-x86_64-bzip2-1.0.8-2
mingw-w64-x86_64-ca-certificates-20210119-1
mingw-w64-x86_64-expat-2.4.3-1 mingw-w64-x86_64-gettext-0.21-3
mingw-w64-x86_64-gmp-6.2.1-3 mingw-w64-x86_64-isl-0.24-1
mingw-w64-x86_64-libffi-3.3-4 mingw-w64-x86_64-libiconv-1.16-2
mingw-w64-x86_64-libsystre-1.0.1-4 mingw-w64-x86_64-libtasn1-4.18.0-1
mingw-w64-x86_64-libtre-git-r128.6fb7206-2 mingw-w64-x86_64-mpc-1.2.1-1
mingw-w64-x86_64-mpdecimal-2.5.1-1 mingw-w64-x86_64-mpfr-4.1.0.p13-1
mingw-w64-x86_64-ncurses-6.3-3 mingw-w64-x86_64-openssl-1.1.1.m-1
mingw-w64-x86_64-p11-kit-0.24.0-1 mingw-w64-x86_64-python-3.9.7-4
mingw-w64-x86_64-readline-8.1.001-1 mingw-w64-x86_64-sqlite3-3.37.2-1
mingw-w64-x86_64-tcl-8.6.11-5 mingw-w64-x86_64-termcap-1.3.1-6
mingw-w64-x86_64-tk-8.6.11.1-2 mingw-w64-x86_64-tzdata-2021e-1
mingw-w64-x86_64-windows-default-manifest-6.4-3
mingw-w64-x86_64-xxhash-0.8.1-1 mingw-w64-x86_64-xz-5.2.5-2
mingw-w64-x86_64-zlib-1.2.11-9 mingw-w64-x86_64-zstd-1.5.1-1
patch-2.7.6-1 pkgconf-1.8.0-1 tar-1.34-2 texinfo-6.8-3
texinfo-tex-6.8-3 base-devel-2022.01-1 mingw-w64-x86_64-binutils-2.37-4
mingw-w64-x86_64-crt-git-9.0.0.6373.5be8fcd83-1
mingw-w64-x86_64-gcc-11.2.0-6 mingw-w64-x86_64-gcc-ada-11.2.0-6
mingw-w64-x86_64-gcc-fortran-11.2.0-6
mingw-w64-x86_64-gcc-libgfortran-11.2.0-6
mingw-w64-x86_64-gcc-libs-11.2.0-6 mingw-w64-x86_64-gcc-objc-11.2.0-6
mingw-w64-x86_64-gdb-11.1-2 mingw-w64-x86_64-gdb-multiarch-11.1-2
mingw-w64-x86_64-headers-git-9.0.0.6373.5be8fcd83-1
mingw-w64-x86_64-libgccjit-11.2.0-6
mingw-w64-x86_64-libmangle-git-9.0.0.6373.5be8fcd83-1
mingw-w64-x86_64-libwinpthread-git-9.0.0.6373.5be8fcd83-1
mingw-w64-x86_64-make-4.3-1 mingw-w64-x86_64-pkgconf-1.8.0-2
mingw-w64-x86_64-tools-git-9.0.0.6373.5be8fcd83-1
mingw-w64-x86_64-winpthreads-git-9.0.0.6373.5be8fcd83-1
mingw-w64-x86_64-winstorecompat-git-9.0.0.6373.5be8fcd83-1

Total Download Size: 163.02 MiB
Total Installed Size: 1073.00 MiB

:: Proceed with installation? [Y/n]
```

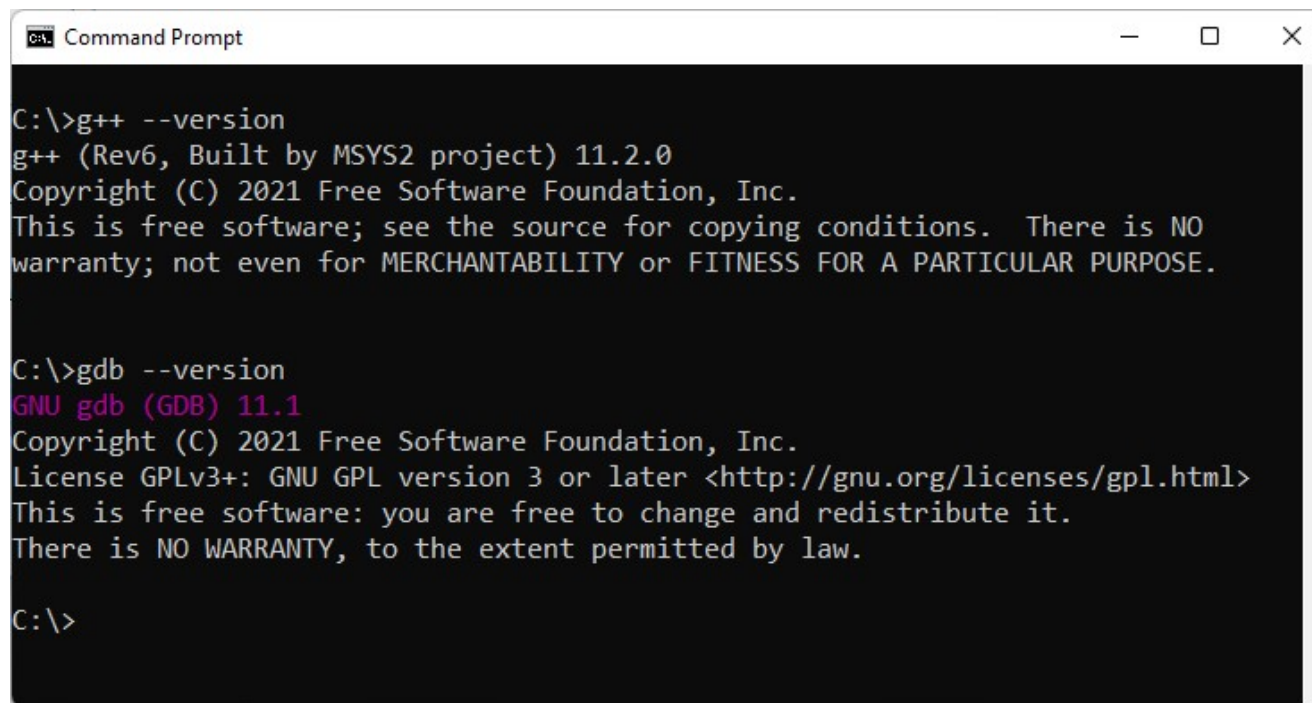
# INSTALAÇÃO

- Adicione no **PATH** do seu usuário o caminho da instalação do MINGW (que você escolheu quando instalou o MSYS2)
  - No Windows 10/11, entre em "Configurações" e procure por "Editar as variáveis de ambiente do sistema".
  - No Windows 7, entre em "Painel de Controle", "Sistema", "Configurações avançadas do sistema", "Propriedades do Sistema", "Variáveis de Ambiente"
  - Se você não alterou o caminho sugerido para instalação do MSYS2, o caminho a ser adicionado da instrução acima na variável **PATH** é:

**C:\msys64\mingw64\bin**

# INSTALAÇÃO

- Teste o **PATH** adicionado e a instalação do compilador C/C++ abrindo um **novo** prompt de comandos:
  - g++ --version
  - gdb --version



```
Command Prompt

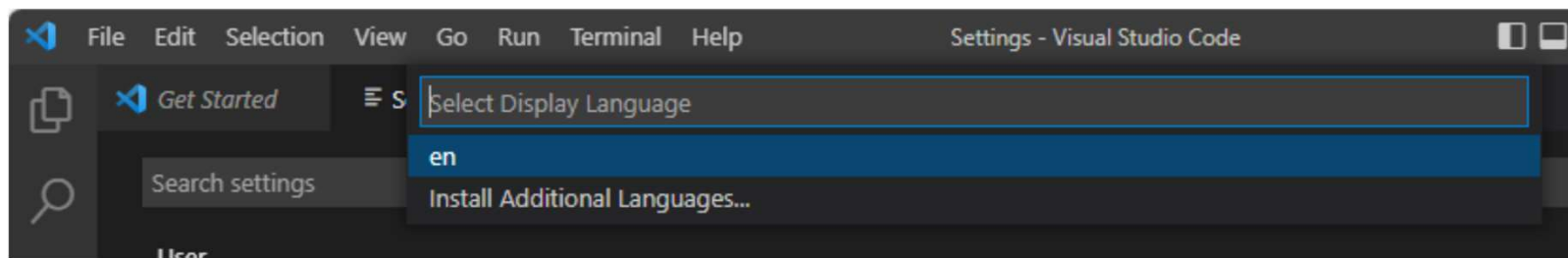
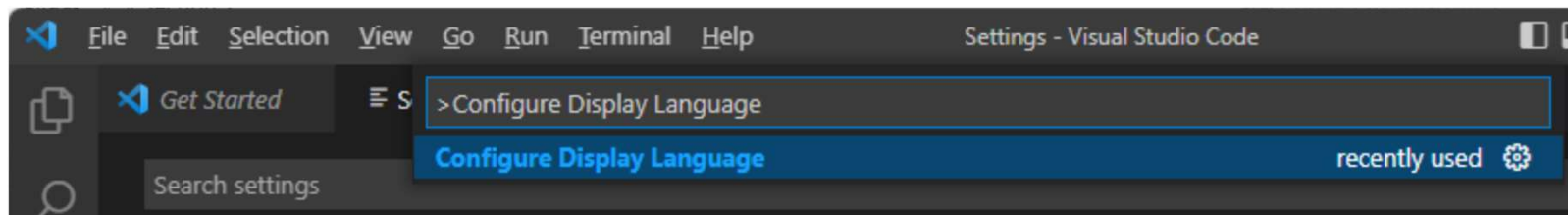
C:\>g++ --version
g++ (Rev6, Built by MSYS2 project) 11.2.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\>gdb --version
GNU gdb (GDB) 11.1
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

C:\>
```

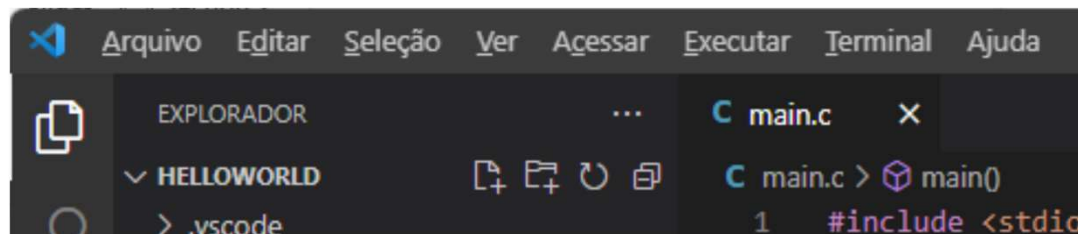
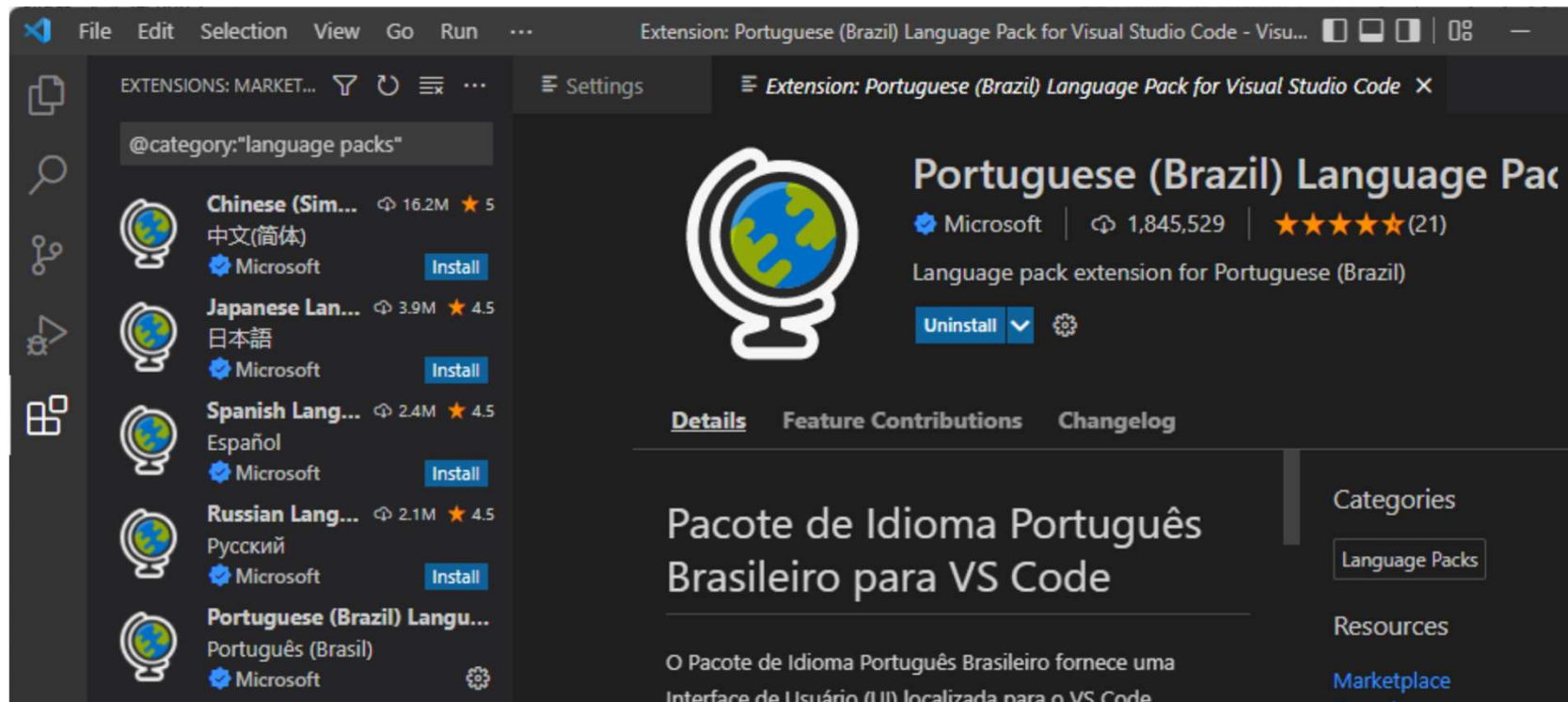
# IDIOMA DA INTERFACE DO VS CODE

- No VS Code, tecle Ctrl + Shift + P para abrir a paleta de comandos e procure por
  - "Configure Display Language" ou
  - "Configurar idioma de exibição"





# IDIOMA DA INTERFACE DO VS CODE



# CRIAÇÃO DE UM PROJETO C

- Em um prompt de comandos:

```
Command Prompt

C:\>mkdir Projetos

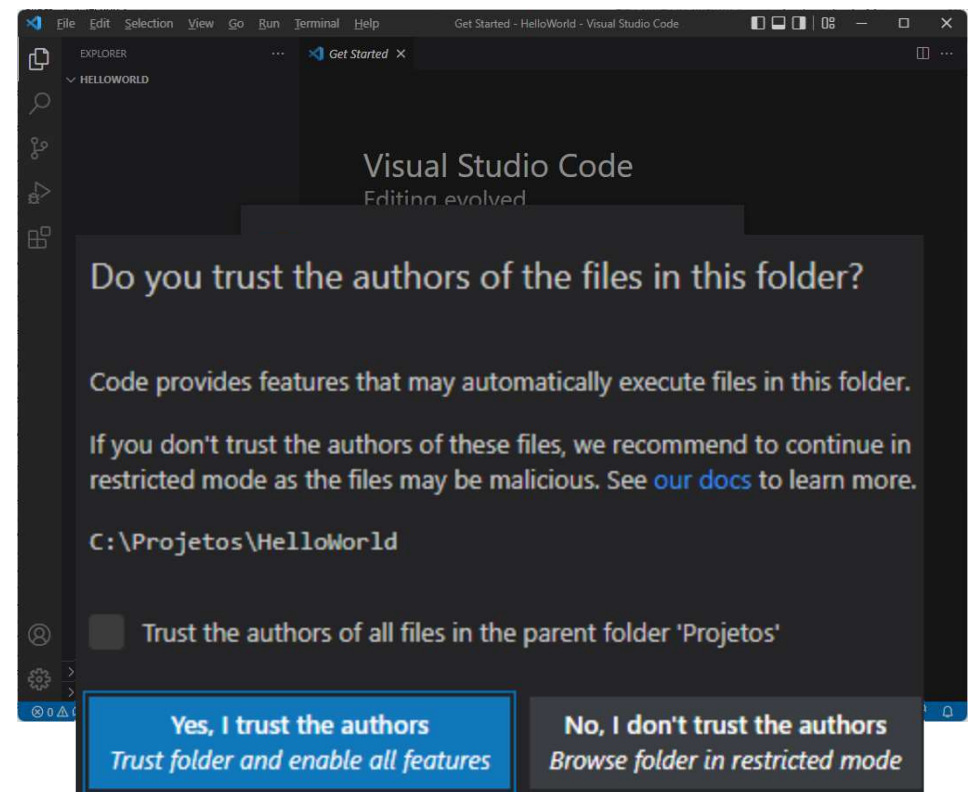
C:\>cd Projetos

C:\Projetos>mkdir HelloWorld

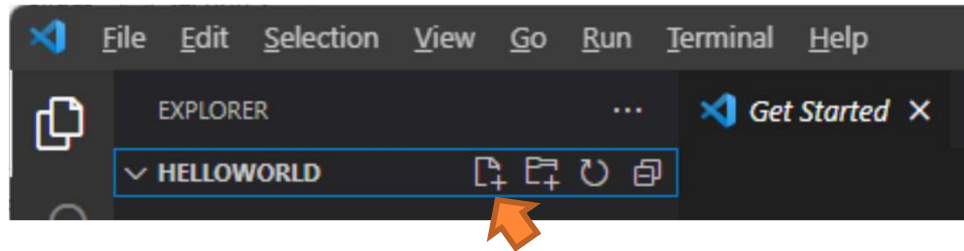
C:\Projetos>cd HelloWorld

C:\Projetos\HelloWorld>code .

C:\Projetos\HelloWorld>_
```

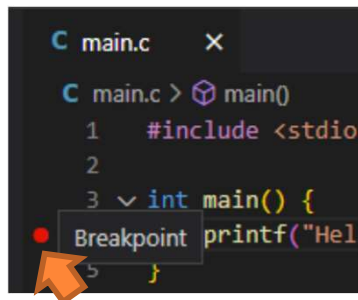


# CRIAÇÃO DE UM PROJETO C



# DEPURAÇÃO / DEBUG

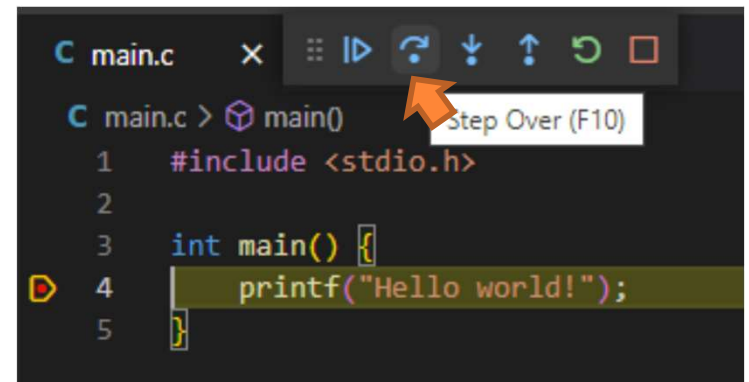
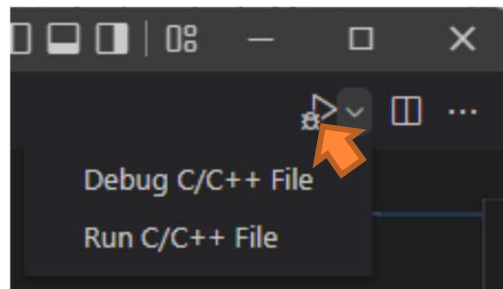
- Indicação de ponto de parada
- Compilação e execução
- Depuração



A screenshot of a code editor window titled 'main.c'. The code is as follows:

```
C main.c > main()
1  #include <stdio.h>
2
3  int main() {
4  printf("Hello world!");
5  }
```

A red dot on line 4 indicates a breakpoint. A tooltip labeled 'Breakpoint' is shown next to it. An orange arrow points to the breakpoint.



A screenshot of a code editor window titled 'main.c'. The code is as follows:

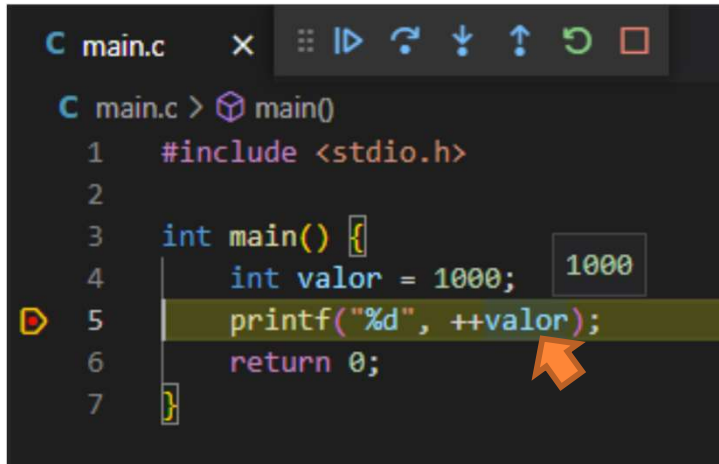
```
C main.c > main()
1  #include <stdio.h>
2
3  int main() {
4  printf("Hello world!");
5  }
```

The line containing 'printf("Hello world!");' is highlighted. A tooltip labeled 'Step Over (F10)' is shown next to the code. An orange arrow points to the 'Step Over (F10)' tooltip.



# DEPURAÇÃO / DEBUG

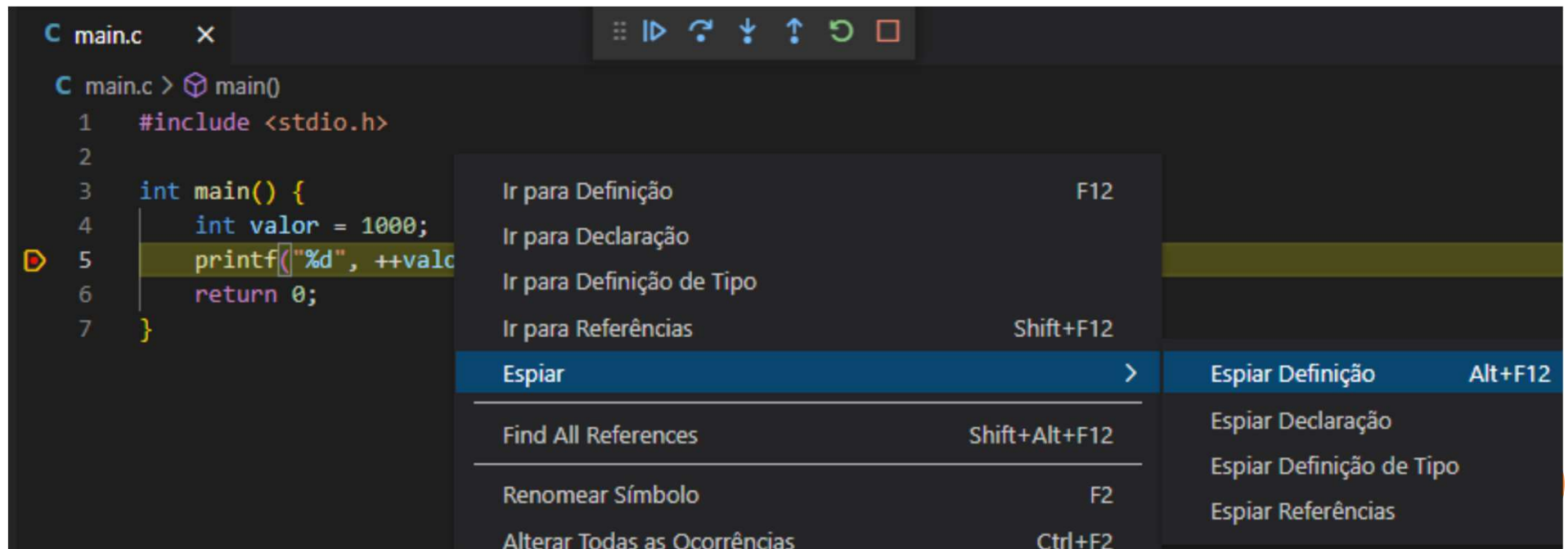
- Durante depuração, ao parar o mouse sobre uma variável, seu valor corrente é apresentado



```
C main.c x [debug icons]
C main.c > main()
1  #include <stdio.h>
2
3  int main() {
4      int valor = 1000;
5      printf("%d", ++valor);
6      return 0;
7  }
```

# DEPURAÇÃO / DEBUG

- Ao clicar com botão direito sobre uma variável, menu "Espiar" permite analisar uso da variável



# REFERÊNCIA

- Microsoft Visual Studio Code + C++ (gcc)
  - <https://code.visualstudio.com/docs/cpp/config-mingw>



The screenshot shows the Visual Studio Code documentation website. The browser address bar displays the URL <https://code.visualstudio.com/docs/cpp/config-mingw>. The page header includes the Visual Studio Code logo and navigation links: Docs, Updates, Blog, API, Extensions, FAQ, and Learn. A search bar labeled 'Search Docs' is on the right. A banner below the header announces 'Version 1.67 is now available! Read about the new features and fixes from April.' The main content area is titled 'Using GCC with MinGW' with an 'Edit' button. A left sidebar lists navigation options: Overview, SETUP, GET STARTED, USER GUIDE, LANGUAGES, NODE.JS / JAVASCRIPT, and TYPESCRIPT. The main text explains that the tutorial configures Visual Studio Code to use the GCC C++ compiler (g++) and GDB debugger from mingw-w64 to create programs that run on Windows. It also mentions that the tutorial does not teach about GCC, GDB, Mingw-w64, or the C++ language, and provides a link to the VS Code documentation repository for further resources.

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn Search Docs

Version 1.67 is now available! Read about the new features and fixes from April.

## Using GCC with MinGW Edit

Overview

SETUP

GET STARTED

USER GUIDE

LANGUAGES

NODE.JS / JAVASCRIPT

TYPESCRIPT

In this tutorial, you configure Visual Studio Code to use the GCC C++ compiler (g++) and GDB debugger from [mingw-w64](#) to create programs that run on Windows.

After configuring VS Code, you will compile and debug a simple Hello World program in VS Code. This tutorial does not teach you about GCC, GDB, Mingw-w64, or the C++ language. For those subjects, there are many good resources available on the Web.

If you have any problems, feel free to file an issue for this tutorial in the [VS Code documentation repository](#).