

Instalación y configuración

Instalar *git* en windows.

- Ve al sitio git-scm.com
- Ingresa en *Install for windows*
- Descarga el último instalador, usando el acceso *Click here to download*
- Ejecuta el instalador, instala utilizando las opciones por defecto

Instalar *Visual Studio Code* en windows.

- Ve al sitio code.visualstudio.com
- Ingresa en *Download for windows*, y la descarga del instalador empezará automáticamente.
- Ejecuta el instalador y sigue las instrucciones

Instalar *MSYS2*

- Ve al sitio www.msys2.org
- En el título *Installation* haz click en el enlace de descarga que comienza por *msys2-x86_64...*
- Ejecuta el instalador e instala msys2 en *C:\msys64*
- Al terminar de instalar, abre una consola de *MSYS2 UCRT64* (si no se abrió al terminar). Si debes abrirla búscala en el menú inicio o en la barra de búsqueda de windows.
- Ejecuta el comando

```
pacman --noconfirm -Syu
```

escribiéndolo en la consola y presionando enter

- Si se cierra la consola, vuelve a abrirla.
- Ejecuta nuevamente

```
pacman --noconfirm -Syu
```

hasta que responda *There is nothing to do* en *Starting core system upgrade...*
y en *Starting full system upgrade...*

Configurar *Visual Studio Code* para usar la terminal de *MSYS2 UCRT64*

- Abre *Visual Studio Code*

- Abre la paleta de comandos con *Control-Mayuscula-P*
- Escribe **new terminal** y selecciona *Create New Terminal (With Profile)*
- Pon el cursor en *bash (MSYS2)* y haz click en el engranaje a su derecha
- Escribe el nombre **bash (MSYS2 UCRT64)** para el nuevo perfil y presiona enter
- Abre nuevamente la paleta de comandos con *Control-Mayuscula-P*
- Escribe **settings** y selecciona *Preferences: Open User Settings (JSON)*
- Busca en el archivo la entrada

```
"bash (MSYS2 UCRT64)": {
  "path": "C:\\msys64\\usr\\bin\\bash.exe",
  "args": [
    "--login",
    "-i"
  ],
  "env": {
    "CHERE_INVOKING": "1"
  }
}
```

- Modificala para que quede de la siguiente manera:

```
"bash (MSYS2 UCRT64)": {
  "path": "C:\\msys64\\usr\\bin\\bash.exe",
  "args": [
    "--login",
    "-i"
  ],
  "env": {
    "MSYSTEM": "UCRT64",
    "CHERE_INVOKING": "1",
    "MSYS2_PATH_TYPE": "inherit"
  }
}
```

- Guarda los cambios y cierra el archivo.
- Abre el archivo `c:\\msys64\\etc\\nsswitch.conf` (con *Control-O* en *Visual Studio Code*)
- Si presenta un diálogo pidiendo confirmación para confiar en el contenido confirma (porque de lo contrario lo abrirá en un modo limitado)

- Busca la línea

```
db_home: cygwin desc
```

- Modifícala para que diga

```
db_home: windows
```

- Guarda los cambios y cierra el archivo
- Abre la paleta de comandos con *Control-Mayuscula-P*
- Escribe

```
terminal profile
```

y selecciona *Terminal: Select Default Profile*

- Selecciona el perfil *bash (MSYS2 UCRT64)*

Crear cuenta en *GitHub*

- Ve al sitio github.com
- Ingresa tu mail y presiona *Sign up for GitHub*
- Sigue las instrucciones. Guarda la contraseña en un lugar seguro

Configurar usuario y email en *git*

- Abre *Visual Studio Code*
- Abre una nueva terminal con el atajo *Control-Mayuscula-Ñ*, el comando *...→Terminal→New Terminal* en la barra de menú o *Terminal: Create New Terminal* en la paleta de comandos
- Ejecuta el comando

```
git config --global user.name "TU NOMBRE"
```

reemplazando *TU NOMBRE* por tu nombre completo

- En el navegador ve a github.com e inicia sesión en tu cuenta si es necesario usando *Sign in*.
- En *GitHub* abre el menú de usuario (en el extremo superior derecho)
- En el menú de usuario selecciona *Settings*

- En el panel izquierdo, en *Access* selecciona *Emails*.
- En *Emails* busca "All web-based Git operations will be linked to " y copia la dirección de mail de la forma `...@users.noreply.github.com` que aparece a continuación
- En *Visual Studio Code*, en la terminal abierta, ejecuta el comando

```
git config --global "...@users.noreply.github.com"
```

. Donde la dirección de correo será la que copiaste en el paso anterior puesta entre comillas sin espacios.

- Puedes comprobar la configuración con el comando

```
git config --global --get-regexp user
```

Instalar las herramientas básicas de desarrollo

- Asegúrate de haber instalado *MSYS2* y configurado *Visual Studio Code* para la terminal *bash* (*MSYS2 UCRT64*)
- Abre *Visual Studio Code*
- Abre una nueva terminal con el atajo *Control-Mayuscula-Ñ*, el comando `...→Terminal→New Terminal` en la barra de menú o *Terminal: Create New Terminal* en la paleta de comandos
- En la terminal ejecuta el comando

```
pacman --noconfirm -S make
```

para instalar *make*

- En la terminal ejecuta el comando

```
pacman --noconfirm -S ${MINGW_PACKAGE_PREFIX}-gcc
```

para instalar el compilador de C para PC

- En la terminal ejecuta el comando

```
pacman --noconfirm -S ${MINGW_PACKAGE_PREFIX}-arm-none-eabi-toolchain
```

para instalar el compilador cruzado.

- En la terminal ejecuta el comando

```
pacman --noconfirm -S ${MINGW_PACKAGE_PREFIX}-openocd
```

para instalar la herramienta de programación y depuración para microcontroladores

- En la terminal ejecutar el comando

```
pacman --noconfirm -S ${MINGW_PACKAGE_PREFIX}-gdb-multiarch
```

para instalar el depurador gdb.

Instalar *Ceedling*

- Asegúrate de haber instalado las herramientas básicas de desarrollo
- Abre *Visual Studio Code*
- Abre una nueva terminal
- Ejecuta el comando

```
pacman --noconfirm -S ${MINGW_PACKAGE_PREFIX}-ruby
```

para instalar el lenguaje de programación *Ruby*, en que está escrito *Ceedling*

- Ejecuta el comando

```
gem install ceedling
```

para instalar *Ceedling*

Instalar *gcovr*

- En un navegador ve a www.python.org/downloads y selecciona *Download Python install manager*
- Descarga y ejecuta el instalador.
- Selecciona *Instalar Python* y procede con la instalación
- Abre *Visual Studio Code*
- Abre una terminal

- Ejecuta el comando

```
py -m pip install gcovr
```