

Contenido del Manual

Contenido

Acerca de Visual Studio Code	2
Requerimientos	2
Instalación y configuración	4
Primer programa en <i>VS Code</i>	6
Uso de una terminal y multiterminales.....	8
Documentación extra	12

Acerca de Visual Studio Code

Visual Studio Code es un IDE (Integrated Development Environment) libre (*Open Source*) que integra al menos las siguientes herramientas:

- Editor de código fuente
- Compiladores
- Interpretes
- Depuradores de código
- Soporte nativo de varios lenguajes de programación como: java, c, php, Python, jsp, html, etc.
- Extensiones para conectar con diferentes herramientas de apoyo.

Esta herramienta es uno de los varios IDE utilizados en la industria para facilitar el desarrollo de código fuente y generar aplicaciones empresariales. Una de las grandes ventajas de utilizar *Visual Studio Code* es la integración de multiterminales de PowerShell, Comand Prompt o Unix Terminal en el mismo espacio de trabajo. Lo anterior facilita el aprendizaje de la compilación, interpretación y ejecución de código a través de las terminales de los sistemas operativos, lo cual otorga flexibilidad en grandes proyectos. En el mismo espacio de trabajo también se puede integrar la herramienta de depuración (*Debugger*), la cual se va ejecutando sobre el editor de texto y a través de una consola iterativa.

Otra de las grandes ventajas, es que, al ser un IDE ampliamente utilizado, cuenta con un gran volumen de documentación técnica en la web y soporte técnico provisto por Microsoft.

Finalmente, es importante mencionar que su GUI (*Graphic User Interface*) es simple de utilizar y da acceso a las diferentes herramientas o extensiones de forma fácil.

Requerimientos

El paquete de instalación se puede encontrar en línea: <https://code.visualstudio.com/docs/setup/setup-overview>

Antes de instalar VS Code se deberá contar con un compilador de C/C++, se recomienda MinGW, el cual se encuentra disponible en línea: <https://sourceforge.net/projects/mingw/files/>

Se deberán instalar los binarios mingw32-base, mingw32-gcc-g++, msys-base en el directorio C:\MinGW

Posteriormente, es necesario agregar la ruta de binarios a las variables de entorno del sistema, en particular en la variable “*Path*”, tal como se muestra la ilustración 1 y 2:

Guía rápida de uso de *Visual Studio Code*

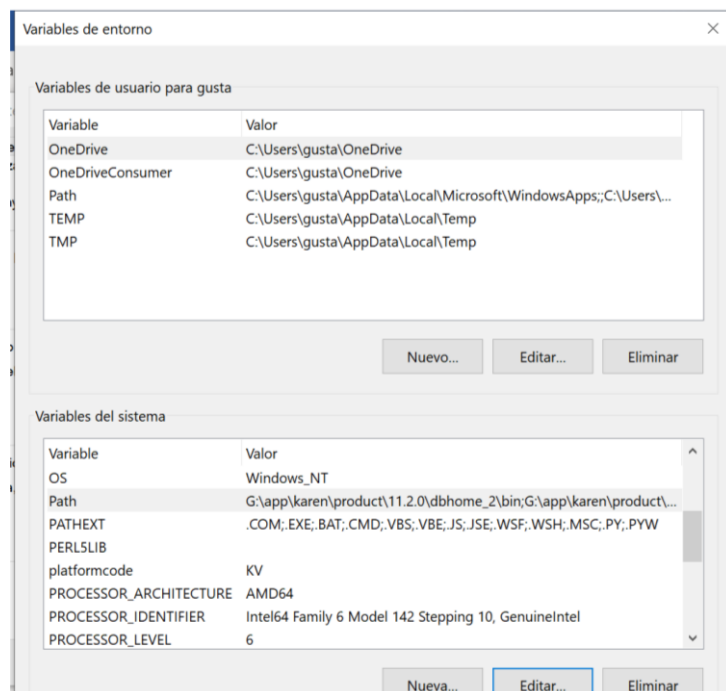


Ilustración 1. Edición de la variable de sistema "Path".

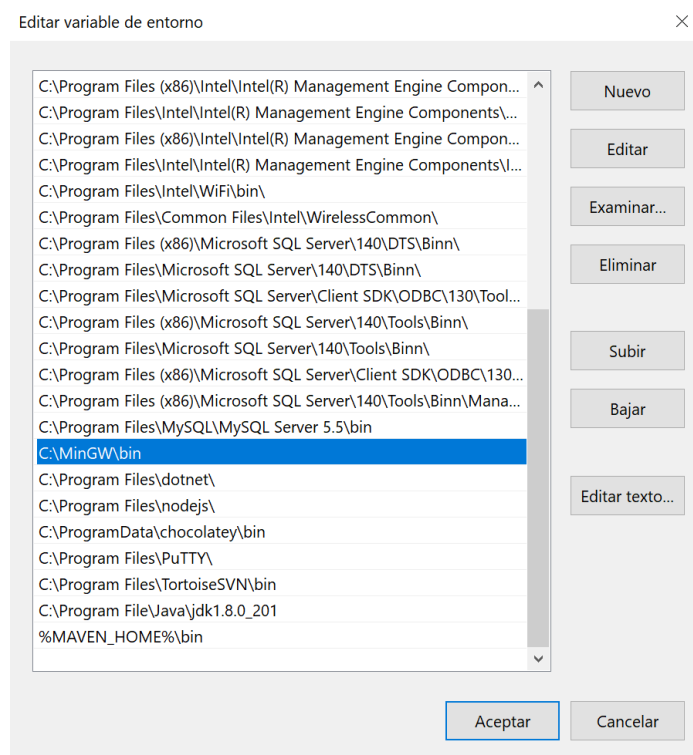


Ilustración 2. Se agrega la ruta de binarios del compilador MinGW a la variable "Path".

Los requerimientos de hardware son:

- 1.6 GHz or faster processor

- 1 GB of RAM

Las plataformas en las cuales se puede instalar VS Code son:

- OS X Yosemite
- Windows 7 (with .NET Framework 4.5.2), 8.0, 8.1 and 10 (32-bit and 64-bit)
- Linux (Debian): Ubuntu Desktop 14.04, Debian 7
- Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 7, Fedora 23

Algunos requerimientos adicionales para la plataforma de Windows son:

- Microsoft .NET Framework 4.5.2

Algunos requerimientos adicionales para la plataforma de Linux son:

- GLIBCXX version 3.4.15 or later
- GLIBC version 2.15 or later

Instalación y configuración

El IDE se puede descargar de la siguiente liga: <https://code.visualstudio.com/>

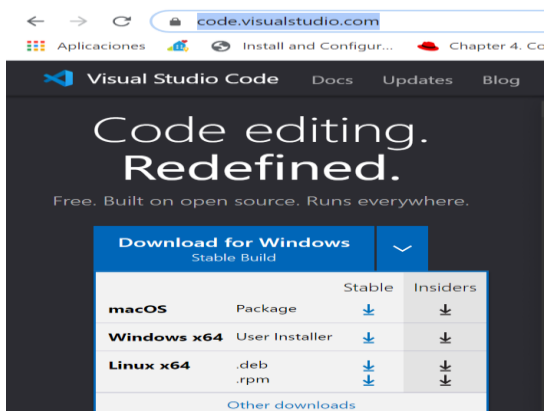


Ilustración 3. Selección de paquete de descarga.

Al descargar el paquete hay que ejecutarlo e instalarlo en la ruta de su preferencia. Al concluir la instalación del IDE deberán ejecutarlo y se mostrará la siguiente ventana:

Guía rápida de uso de *Visual Studio Code*

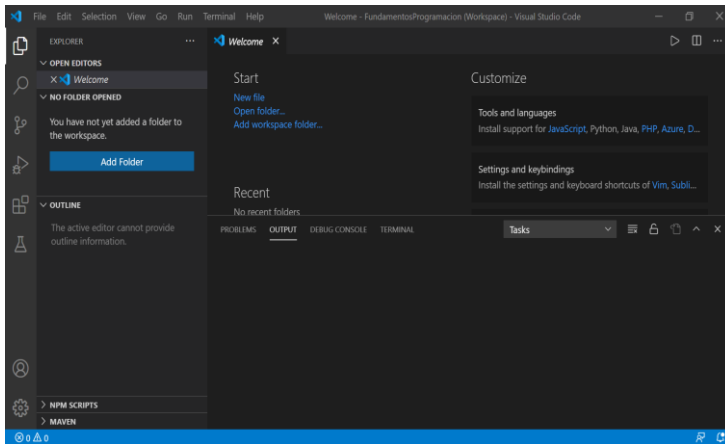


Ilustración 4. Página principal de VS Code

Luego, hay que instalar las siguientes extensiones:

- C/C++ IntelliSense, debugging, and code browsing.
- Compile & Run single c/c++ files easily

Esto se logra dando clic en la sección de extensiones tal como se muestra en la siguiente imagen:

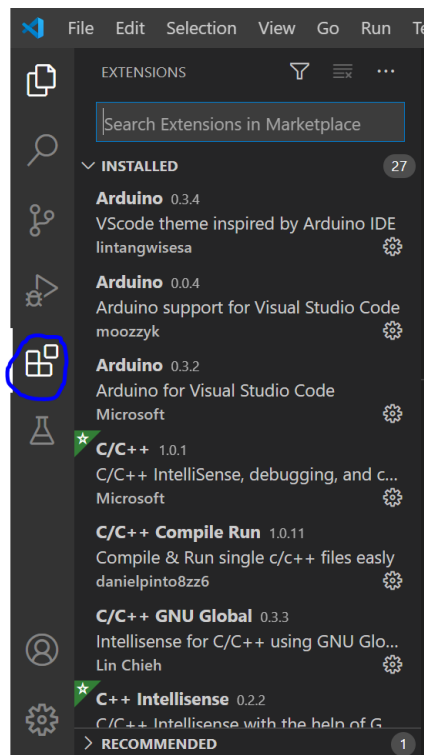


Ilustración 5. Sección de extensiones.

Se deberán buscar las extensiones e instalar dando clic en el botón de “Install”, como se muestra a continuación:

Guía rápida de uso de *Visual Studio Code*

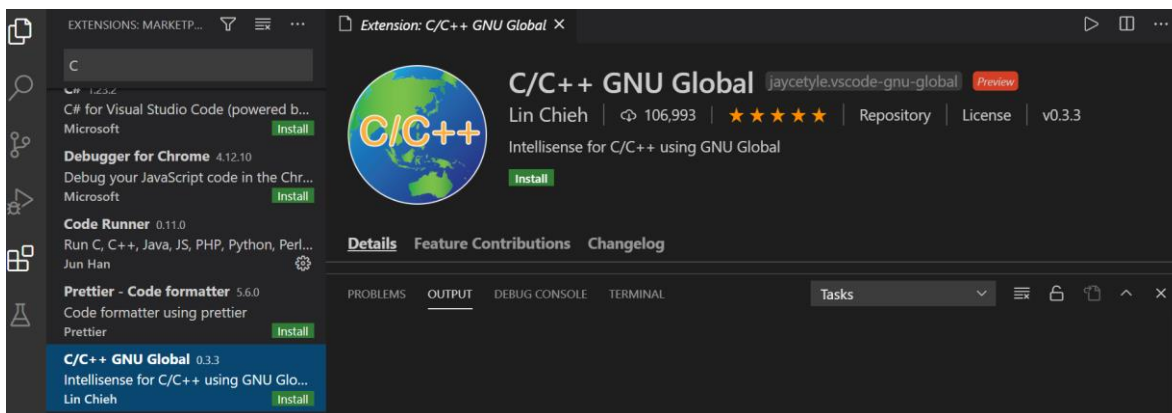


Ilustración 6. Búsqueda de extensiones.

Al concluir la instalación de las extensiones básicas debemos regresar a la sección “*explorer*” para poder crear nuestro primer proyecto.

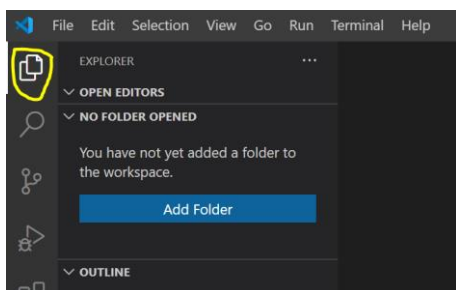


Ilustración 7. Sección “*explorer*”

Ya ubicados en dicha sección entonces se deberá agregar un folder para resguardar archivos fuente y ejecutables, para esto deberán dar clic en el botón “*Add Folder*” y seleccionar la carpeta donde deseen ubicar dicha información. Después de haber seleccionado la carpeta el IDE lo reconocerá como su *workspace* y podrán observar su contenido:

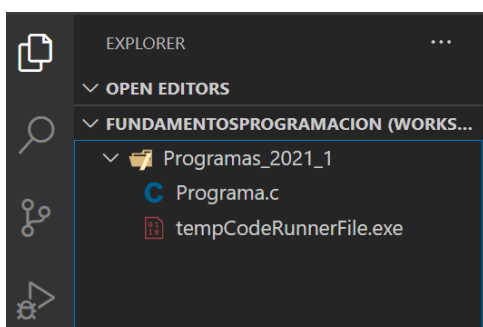


Ilustración 8. Contenido *workspace*

Finalmente, ya configurado su *workspace* se puede comenzar a trabajar en el primer programa.

Primer programa en *VS Code*

El primer programa que se realizará es un hola mundo en el lenguaje C, y se realiza con el siguiente código:

```
#include<stdio.h>

int main(){

    printf("¡Hola mundo!");

    return 0;

}
```

Primeramente, hay que crear un nuevo archivo, para eso deberán dar clic en el botón “File/New File” tal como se muestra en la siguiente ilustración:

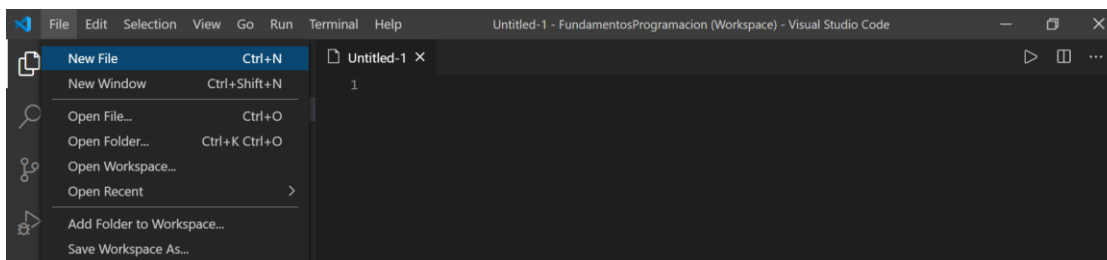


Ilustración 9. Creación de nuevo archivo.

Después de haber creado el archivo, deberán guardarlo en su *workspace*, para eso deberán dar clic en “File/Save As...” y asignarle un nombre a su archivo con la extensión *.c*, en este ejemplo lo guardamos como “Hola.c”:

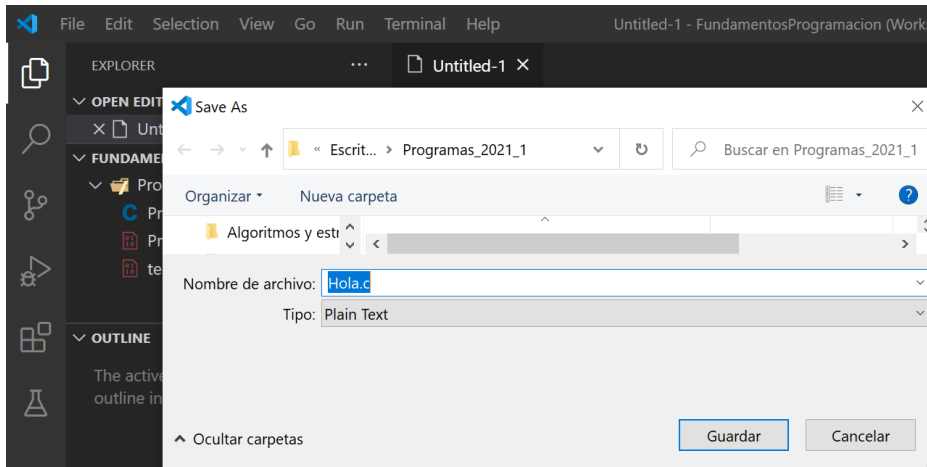


Ilustración 10. Proceso de guardado de nuevo archivo.

Luego, ya que se guardó el documento éste se verá reflejado en el *workspace*. Y, en el espacio “Open Editors” se mostrará un icono de una “C” que identifica que el archivo pertenece a un programa descrito en lenguaje C.

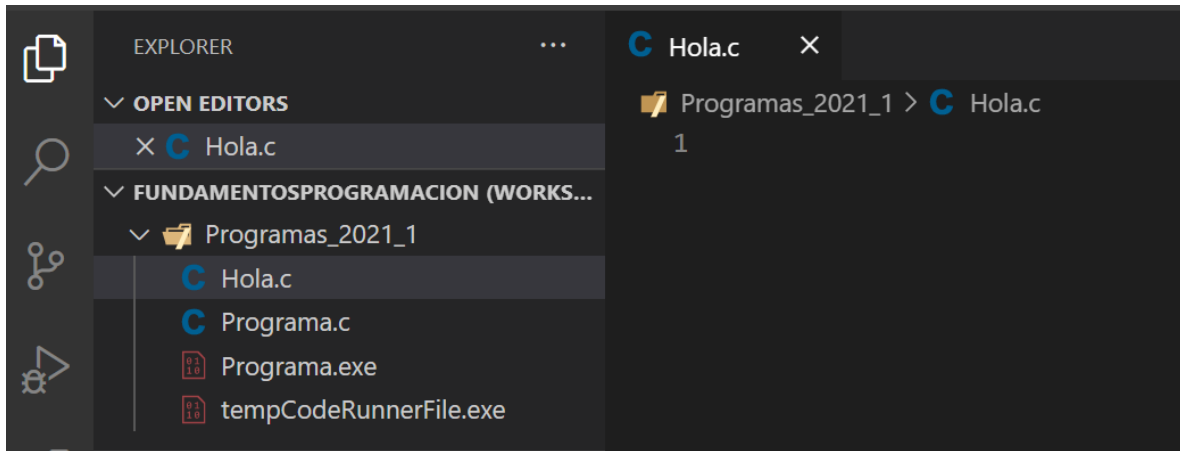


Ilustración 11. Muestra después del guardado.

Ahora se debe transcribir el código al archivo “.c” para posteriormente compilarlo y ejecutarlo. En el IDE hay que identificar dos secciones (marca roja y azul) que se muestran en la Ilustración 10. El botón marcado con color azul es “play” y sirve para compilar y ejecutar el programa descrito en el archivo abierto. La sección roja indica un panel formado por 4 secciones y cada una cuenta con una función que se describen de forma general como sigue:

- **PROBLEMS:** Aquí podrán revisar los posibles errores de sintaxis y semántica que pudiera tener los archivos fuente.
- **OUTPUT:** Aquí se muestran los resultados de la ejecución de un programa, siempre y cuando se haya ejecutado a través del botón de “play”
- **DEBUG CONSOLE:** Aquí se muestran los resultados de ejecución paso a paso resultado de un proceso de *Debugging*. Para esto es necesario contar con la extensión para Visual Studio Code para el lenguaje con el que se esté trabajando (disponibles en la sección de extensiones)
- **TERMINAL:** Es el lugar donde se puede llamar a una terminal como un PowerShell, CMD, etc. Con la finalidad de apoyar a la compilación o ejecución de programas.

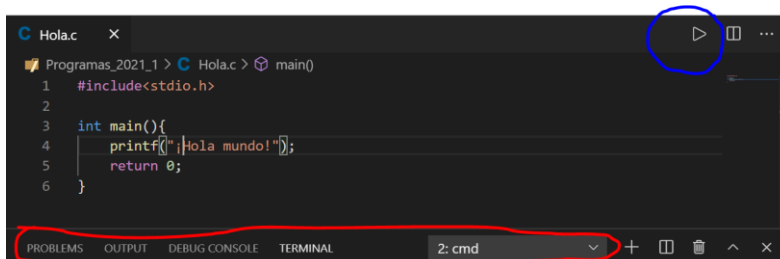


Ilustración 12. Secciones importantes del IDE

Uso de una terminal y multiterminales

Una importante característica de este IDE es la capacidad de agregar multiterminales de forma simple y con diversos sets de comandos (Unix/Windows). A continuación, utilizaremos comandos Windows en una terminal **cmd** para compilar y ejecutar el programa que anteriormente describieron:

Guía rápida de uso de *Visual Studio Code*

- `cd path → cd C:\Users\gusta\Desktop\Programas_2021_1` (Se busca el archivo fuente)
- `gcc File.C -o File.exe → gcc Hola.c -o Hola.exe` (Se compila el archivo fuente)
- `File.exe → Hola.exe` (Se ejecuta el programa)

En la siguiente ilustración se muestran los comandos ejecutándose en el IDE.

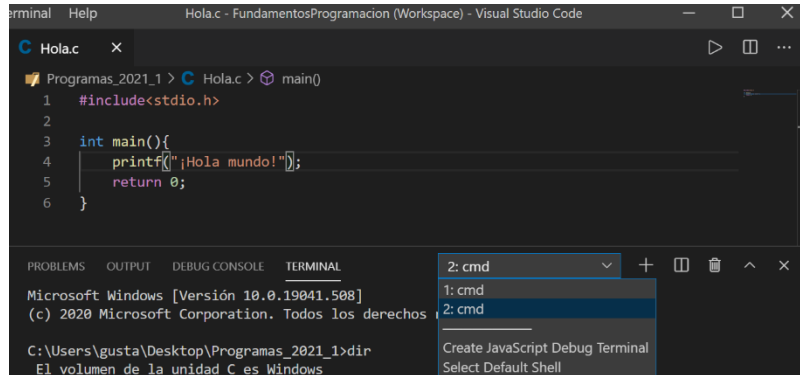


Ilustración 14. Selección de terminales

Duplicidad de archivos y terminales es una de las características más útiles del IDE, ya que dicha funcionalidad permite tener dos versiones del mismo archivo para dar mejor seguimiento a cualquier programa o a comandos ejecutados con anterioridad. La funcionalidad se aplica cuando se da clic en el botón de dos hojas (ver círculo rojo).

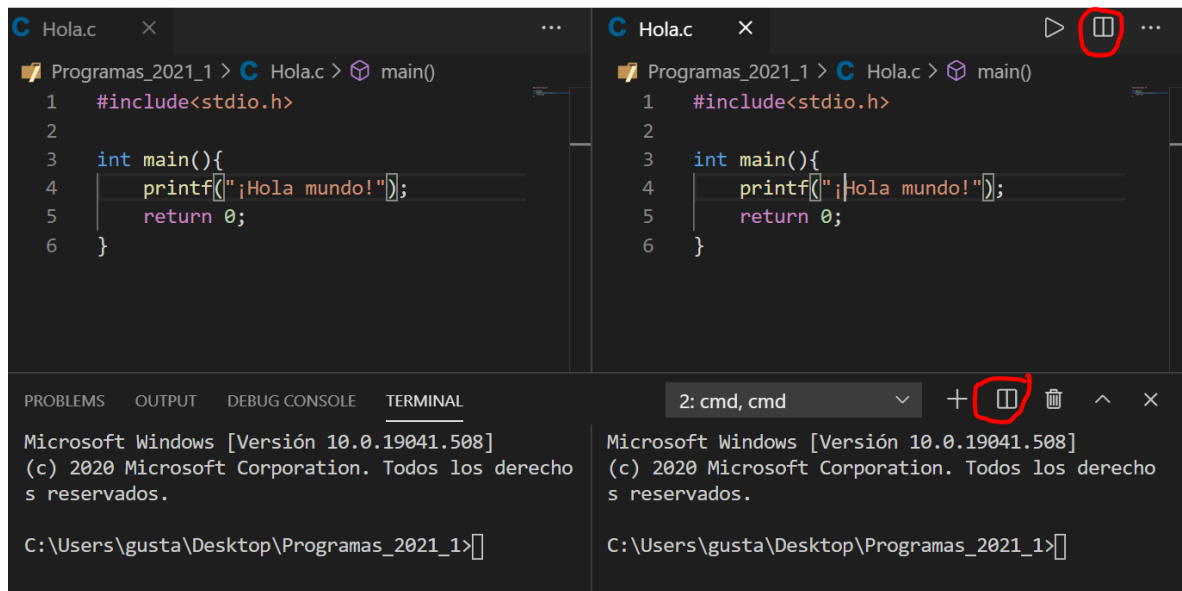


Ilustración 15. Duplicidad de archivos y terminales.

Instalación de R

Para poder trabajar con el lenguaje R sobre VS Code será necesario descargar e instalar los paquetes binarios de R, disponibles en: <https://cran.r-project.org/>

La versión a descargar es la 4.0.4.

El segundo paso es instalar R en la siguiente ruta: C:\Program Files\R\R-4.0.4

Cuando el instalador termine entonces deberán modificar las variables de entorno de su equipo de computo para agregar la ruta “C:\Program Files\R\R-4.0.4\bin” en la variable del sistema “Path”, tal como se observa en las siguientes ilustraciones:

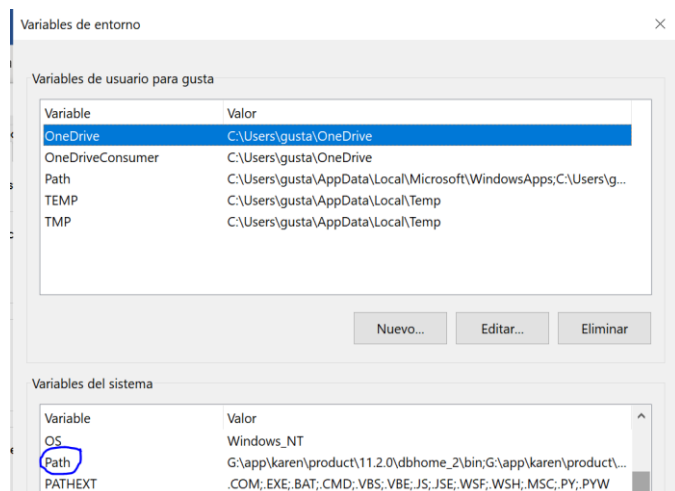


Ilustración 16. Variable Path.

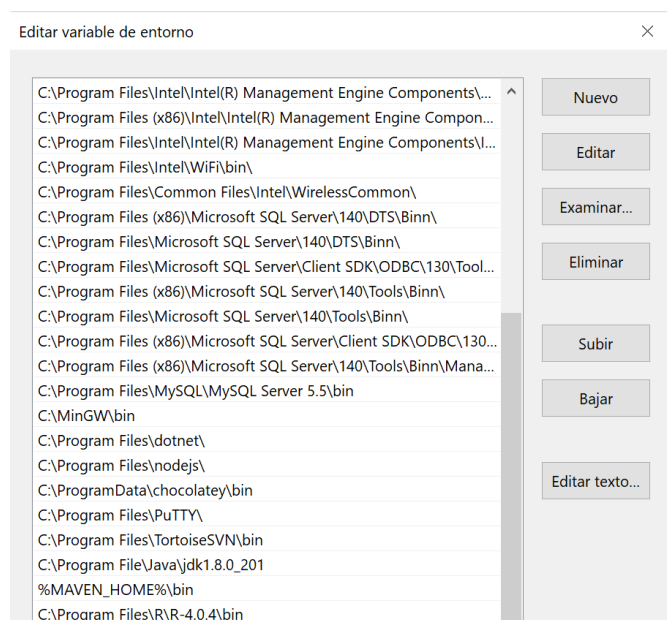
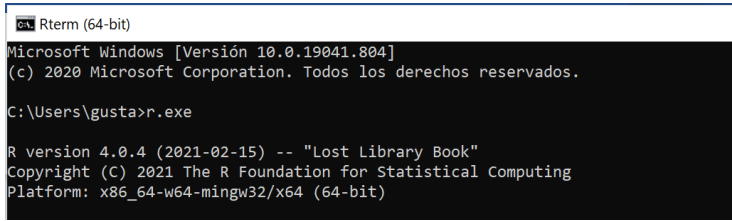


Ilustración 17. Modificación de variable Path.

Para validar que se guardó la ruta correctamente se deberá abrir una terminal CMD y escribir el comando “r.exe” tal como se observa en la siguiente ilustración:



```
Rterm (64-bit)
Microsoft Windows [Versión 10.0.19041.804]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\gusta>r.exe

R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (c) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

Ilustración 18. Ejecución de terminal R.

En la terminal que se está ejecutando de R, se debe escribir el siguiente comando:

- `install.packages("languageserver")`

En la ventana emergente “Secure CRAN” seleccionen “México https” y esperan a que concluya la instalación de los paquetes adicionales.

Después de haber instalado R y haber concluido la instalación del “languageserver” procederemos a instalar las siguientes extensiones en VS Code:

- VSCode R Extension by Yuki Ueda
- R LSP Client by REditorSupport

Luego, se deberán instalar el módulo “Radian” de Python (se deberá tener de antemano instalado Python 2.7, 3.7 o 3.8), a través del siguiente comando “`pip install -U radian`”.

Finalmente, hay que agregar las siguientes líneas al archivo “settings.json”, el cual se encuentra en “`.\Microsoft VS Code\resources\app\extensions\emmet\node_modules\vscode-emmet-helper\node_modules\vscode-uri\vscodes\`”:

```
{
  "r.bracketedPaste": true,
  "r.terminal.windows": "C:\\Python38\\Scripts\\radian.exe",
  "r.lsp.path": "C:\\Program Files\\R\\R-4.0.4\\bin\\R.exe",
  "r.lsp.debug": true,
  "r.lsp.diagnostics": true,
  "r.terminal.option": [
    "--no-save",
    "--no-restore",
    "--r-binary=C:\\Program Files\\R\\R-4.0.4\\bin\\R.exe"
  ],
}
```

Hay que modificar la ruta para Python, dependerá de donde lo tengan instalado para la siguiente variable:

- `r.terminal.windows`

Para confirmar que los archivos “.r” se podrán ejecutar en VS Code bastará con escribir el siguiente código en un archivo “prueba.r” y darle run (botón de *play*) tal como se observa en la siguiente ilustración:

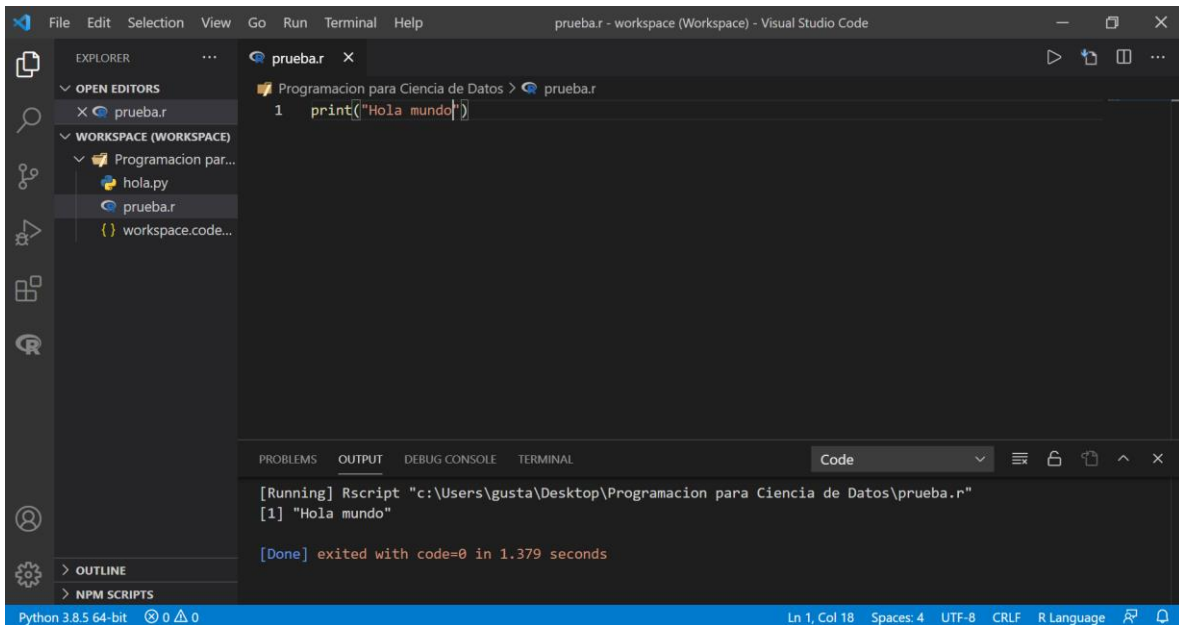


Ilustración 19. Hola mundo en R.

Documentación extra

Se puede encontrar más sobre esta herramienta en los siguientes documentos online:

- Mas sobre la instalación: <https://code.visualstudio.com/docs/setup/windows>
- Configuración para cada plataforma: <https://code.visualstudio.com/docs/setup/setup-overview>
- Guía de usuario: <https://code.visualstudio.com/docs/editor/codebasics>
- Soporte a lenguaje C y C++: <https://code.visualstudio.com/docs/languages/cpp>
- Proceso de evaluación de código(debugging): <https://code.visualstudio.com/docs/editor/debugging>
- Control de versiones: <https://code.visualstudio.com/docs/editor/versioncontrol>
- Multiproyectos: <https://code.visualstudio.com/docs/editor/multi-root-workspaces>
- Más sobre extensiones: <https://code.visualstudio.com/docs/introvideos/extend>
- Instalación de R: <https://medium.com/analytics-vidhya/a-fresh-start-for-r-in-vscode-ec61ed108cf6>

- Instalación de Python en VS Code: <https://code.visualstudio.com/docs/languages/python>