



Lenguajes y Autómatas II

Nombre del equipo: INTERNET EXPLORER 95

Nombre de integrante: Rodríguez Perez Luis diego
no control: 161080170

nombre de integrante: Ramírez Peña Carlos Iván
no control: 161080215

nombre de integrante: Hernández Ruiz Adrián Felipe
no control: 171080105

nombre de integrante: Álvarez Rosales Alejandro
no control: 151080010

Grupo: 7 AM

NOMBRE DE LA ESCUELA: Instituto Tecnológico de Iztapalapa

Carrera: Ing. en Sistemas computacionales

Nombre de la materia: Lenguajes y Autómatas 2

Nombre del Profesor: Parra Hernández Abiel

tema: Sub proyecto. - Clang: a C language family frontend for end

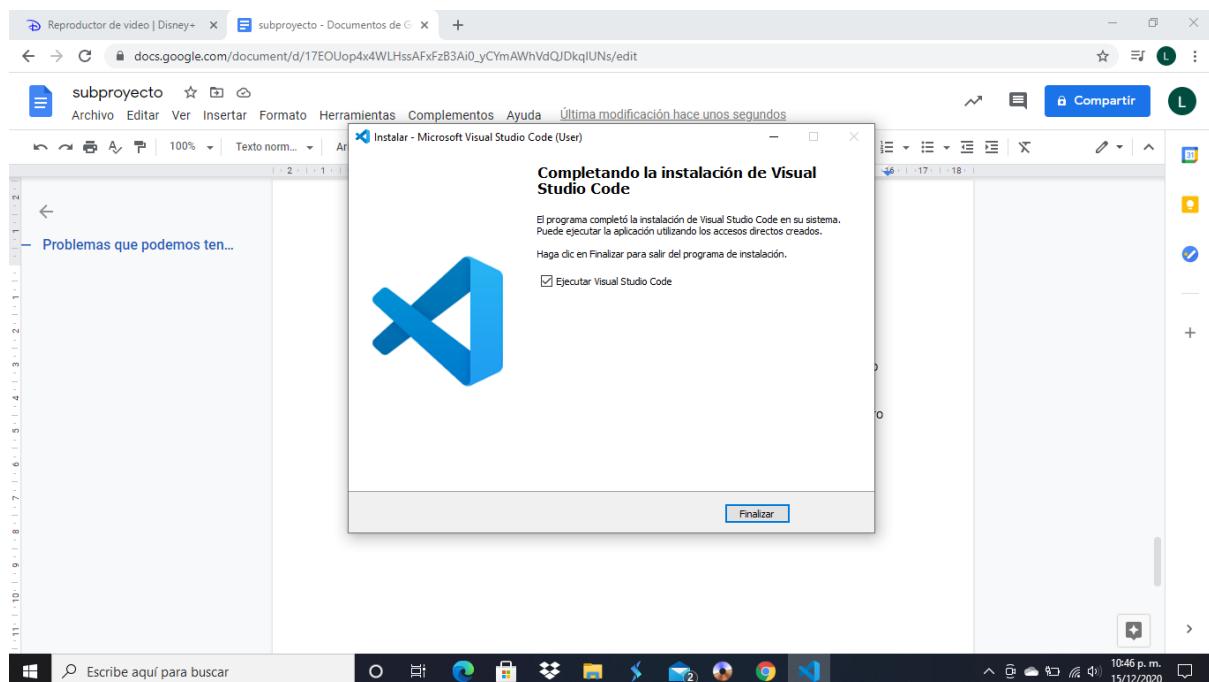


Problemas que podemos tener al empezar en el proyecto

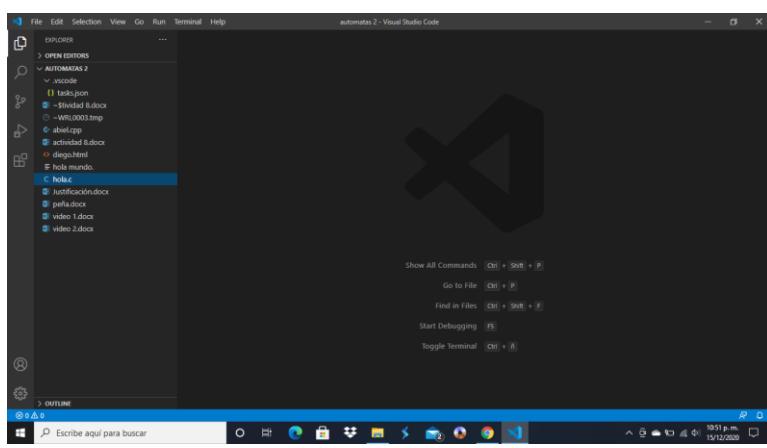
Problemas que podemos tener al correr nuestro código de c

para empezar antes de ver los problemas vamos a empezar ver como funciona y qué es lo que vamos a utilizar en este proyecto

vamos a empezar a utilizar nuestro visual studio code primero tenemos que instalar nuestro visual el visual debe de ser de 64 bits para poder trabajar bien



ya que está instalado vamos a abrirlo para poder trabajar



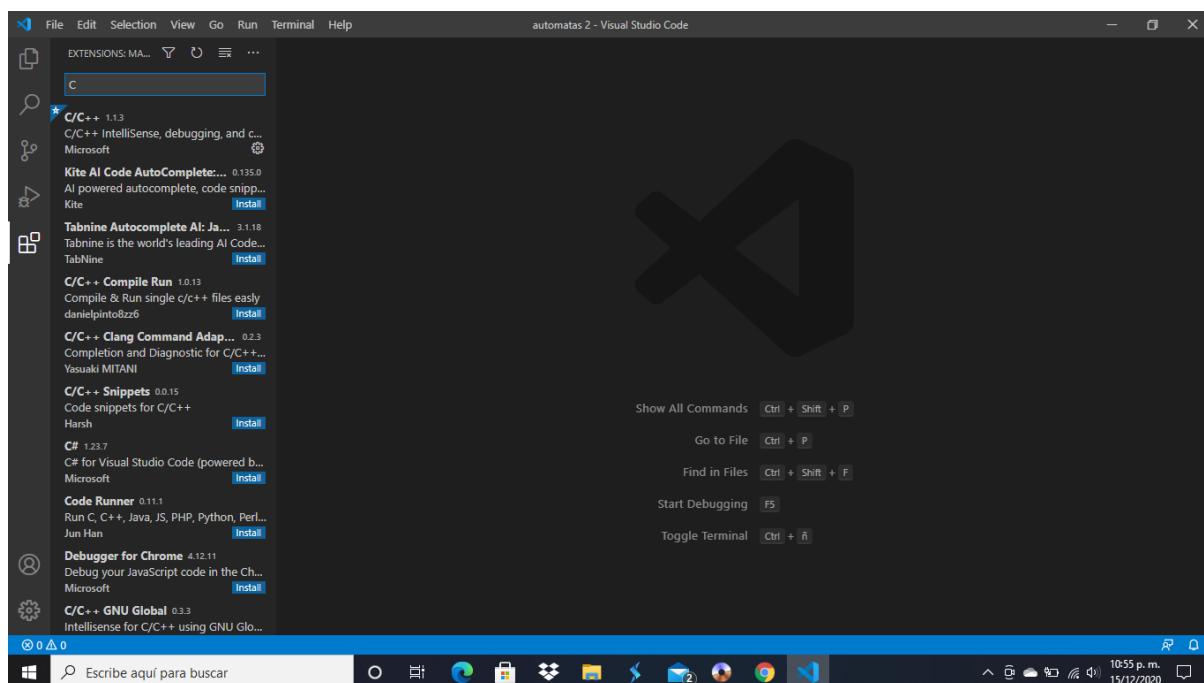


ya que lo abrimos vamos a ver qué herramientas vamos a utilizar para poder crear nuestro programa en visual nos dan las extensiones que vamos a utilizar a qué me refiero con extensiones son los lenguajes de programación que vamos a ocupar que son:

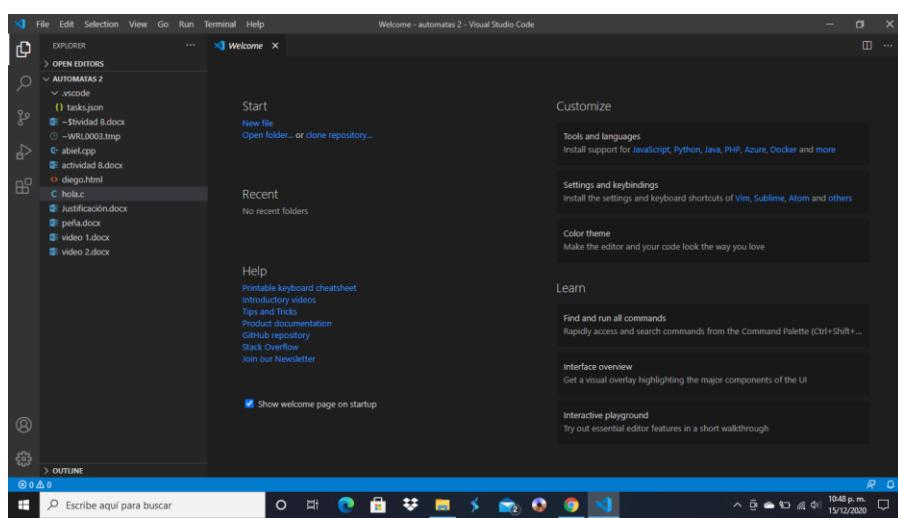
1.- C\C++

2.-C\C++ COMPILE RUN

son los que vamos a utilizar



los instalamos ya que los tengamos instalados vamos a crear nuestro programa creamos uno nuevo en c por ejemplo vamos a hacer un pequeño programa que es un hola mundo a ver que sucede





como podemos ver ya tengo una carpeta que se llama autómatas 2 donde vamos a estar metiendo nuestros programas para hacer nuestro proyecto

```
File Edit Selection View Go Run Terminal Help
EXPLORER OPEN EDITORS
AUTOMATAS 2 vscode tasks.json ~$trividad 8.docx -WRL0003.tmp abiel.cpp actividad 8.docx diego.html hola mundo hola.c Justificación.docx peña.docx video 1.docx video 2.docx
holo mundo. C hola.c x
holo mundo > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main () {
5
6     printf("hola mundo\n");
7     system("pause");
8     return 0;
9 }
```

Escribe aquí para buscar

antes que veamos lo de nuestro programa vamos a ver los ahora sí que las propiedades que podemos ver cuales lenguajes nuestro editor de texto, las extensiones y varias cosas

File Edit Selection View Go Run Terminal Help

Settings - automatas 2 - Visual Studio Code

EXPLORER OPEN EDITORS
AUTOMATAS 2 vscode tasks.json ~\$trividad 8.docx -WRL0003.tmp abiel.cpp actividad 8.docx diego.html hola mundo hola.c Justificación.docx peña.docx video 1.docx video 2.docx
Turn on Settings Sync

Commonly Used

Commonly Used

Files: Auto Save

Controls auto save of dirty editors. Read more about autosave [here](#).

off

Editor: Font Size

Controls the font size in pixels.

20

Editor: Font Family

Controls the font family.

Consolas, 'Courier New', monospace

Editor: Tab Size

The number of spaces a tab is equal to. This setting is overridden based on the file contents when [Editor: Detect Indentation](#) is on.

4

Editor: Render Whitespace

Controls how the editor should render whitespace characters.

Escribe aquí para buscar

ahora si nos pasamos a nuestro código de hola mundo



ya que tenemos nuestro código vamos a correrlo a ver qué es lo que pasa

holo.c - automatas 2 - Visual Studio Code

```
C hola.c x
C hola.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (){
5
6     printf("holo mundo\n");
7     system("pause");
8     return 0;
9 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

> Executing task: C/C++: cl.exe build active file <

Starting build...

Build finished with errors(s):

"cl.exe" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

The terminal process terminated with exit code: -1.

Terminal will be reused by tasks, press any key to close it.

Ln 9, Col 2 Spaces: 4 UTF-8 CRLF C Win32 R D

Escribe aquí para buscar

11:04 p. m. 15/12/2020

prácticamente nos marca eso hay un error no en el código si no encuentra un compilador para que pueda correr nuestro programa pues más bien el visual son para que podamos crear, pero es muy complicado correr nuestro programa ahora sí que es un problema que podemos tener en este proyecto

pero ahora estuvimos investigando cómo podemos arreglar ese problema podemos utilizar un compilador llamado MINGW ahí es donde podemos correr nuestro programa

primero vamos a descargar nuestro MINGW de 64 bits para nuestra computadora lo buscamos en el buscador y nos vamos a la primera página que es esta

Reproductor de video | Disney+ | subprojeto - Documentos de Google | mingw - Buscar con Google

Google

mingw

Cerca de 2,770,000 resultados (0.39 segundos)

http://www.mingw.org ▾ Traducir esta página

MinGW | Minimalist GNU for Windows

MinGW compilers provide access to the functionality of the Microsoft C runtime and some language-specific runtimes. MinGW, being Minimalist, does not and ...

HOWTO Install the MinGW. Welcome to MinGW.org - Install MinGW - Getting Started

https://sourceforge.net › projects ▾ Traducir esta página

MinGW - Minimalist GNU for Windows download ...

MinGW: A native Windows port of the GNU Compiler Collection (GCC), with freely distributable import libraries and header files for building native Windows

★★★★☆ Calificación: 4.1 - 177 votos · Gratis · Windows · Utilidades/Herramientas

Mingw-get · MinGW - Minimalist GNU for ... · Mingw-get-inst · Mingw-get-0.4-alpha-1

https://sourceforge.net › projects ▾ Traducir esta página

MinGW-w64 - for 32 and 64 bit Windows download ...

The mingw-w64 project is a complete runtime environment for gcc to support binaries native to Windows 64-bit and 32-bit operating systems. Features: Compiler ...

★★★★★ Calificación: 4.4 - 109 votos · Gratis · Windows

Escribe aquí para buscar

ya que le damos clic y nos mandara a la página de MINGW y nos vamos a descargas

The screenshot shows the official website for MinGW (GNU minimalist para Windows). The main content area features a welcome message and information about the project's history and purpose. On the left, there's a sidebar with links for administrative updates, user session initiation, and navigation. A search bar is at the top. The right side includes sections for user search, recent releases, and a detailed list of available packages.

ya que vamos que le dimos en descargas ya nos manda una lista donde esta los paquetes de MINGW y le damos click en el simbolo de windows

The screenshot shows the download page for the mingw-get-setup.exe file from osdn.net. The page displays the file's details, including its rating (3.9 stars), number of reviews (54), and download link. Below this, a list of available packages is shown, each with a download link. A cookie consent banner is visible at the bottom of the page.

y automáticamente se va descargando nuestro MINGW



Reproductor de video | Disney+ | subprojeto - Documentos de Google | Downloading File /68260/mingw-get-setup.exe | +

osdn.net/projects/mingw/downloads/68260/mingw-get-setup.exe/

File Information

File Size: 93,184 bytes
MD5: 60737d3eabafc0def38e26fa31648b35
SHA1: 997a8f7136da1ae6c188b98aab604229355f5a7
SHA256: a5de8c70676ffd157414e3116c7cb63dba48e549b5880177b10e4e6bc6c282e

Virus Check: Uploaded files are scanned with VirusTotal on OSDN.

Where do you want to go next?

Go to the Web page of MinGW - Minimalist GNU for Windows project | Go to the project page on OSDN | View another version

Project Description

This is the official download site for the latest packages originating from the [MinGW.org Project](#).
MinGW is a native Windows port of the GNU Compiler Collection (GCC), with freely distributable import libraries and header files for building native Windows applications; includes extensions to the MSVC runtime

Review

Average: 3.9

Star Rating	Count
5 Stars	27
4 Stars	5
3 Stars	17

mingw-get-setup.exe

Escribe aquí para buscar

11:20 p.m. 15/12/2020

y lo instalamos lo vamos a guardar en nuestro disco c para que esté ahí nuestro ming

Reproductor de video | Disney+ | subprojeto - Documentos de Google | Downloading File /68260/mingw-get-setup.exe | +

docs.google.com/document/d/17EOUop4x4WLhsAFxFzBAl0_yCymAWhVdQJdkqIUNs/edit

subprojeto

Archivo Editar Ver Insertar Formato Complementos Ayuda Última modificación hace unos segundos

MinGW Installation Manager Setup Tool

mingw-get version 0.6.3-pre-20170905-1

Step 2: Download and Set Up MinGW Installation Manager

Connecting to osdn.net ... | 262.49 kB of 262.49 kB : 100 %

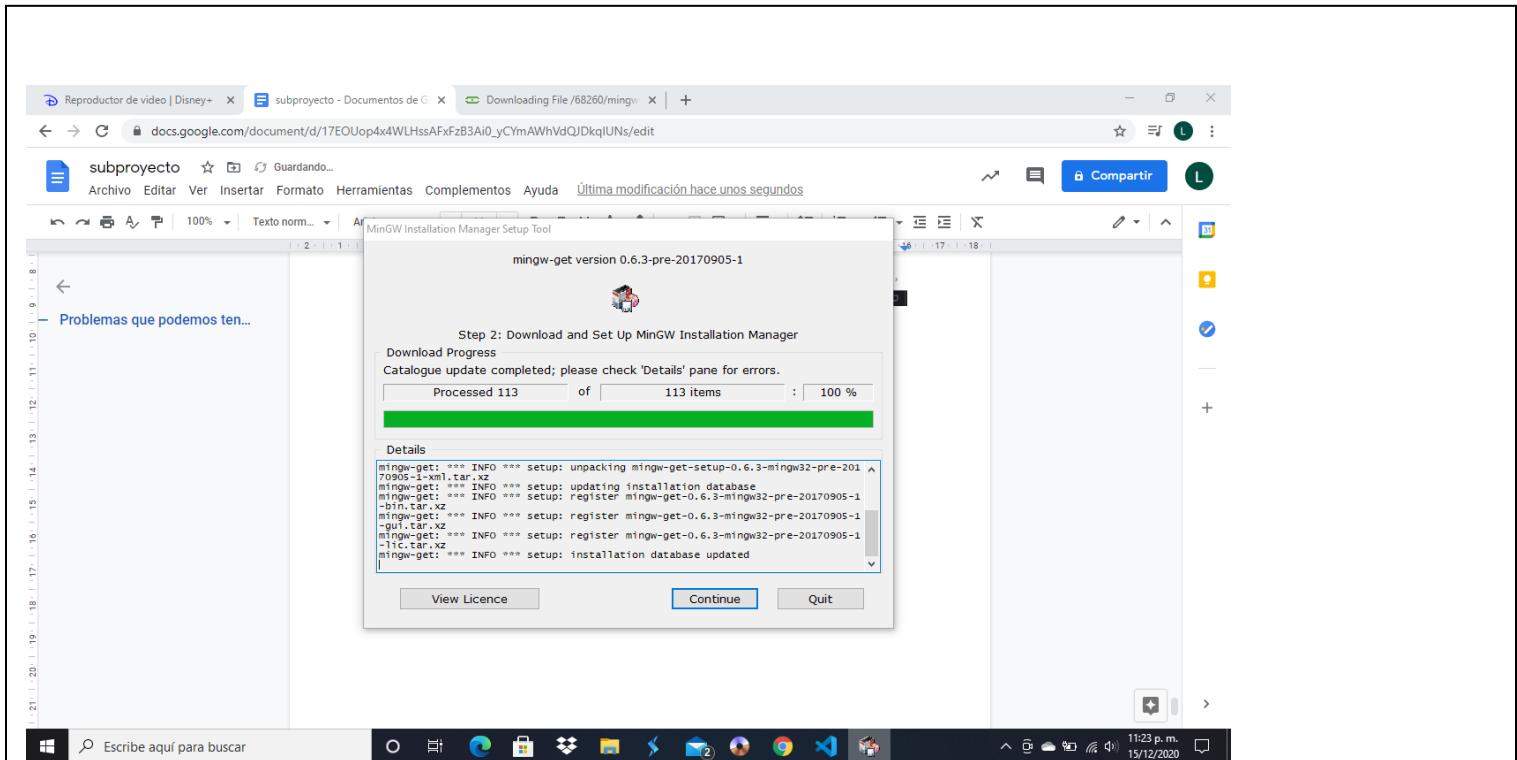
Details

View Licence Continue Quit

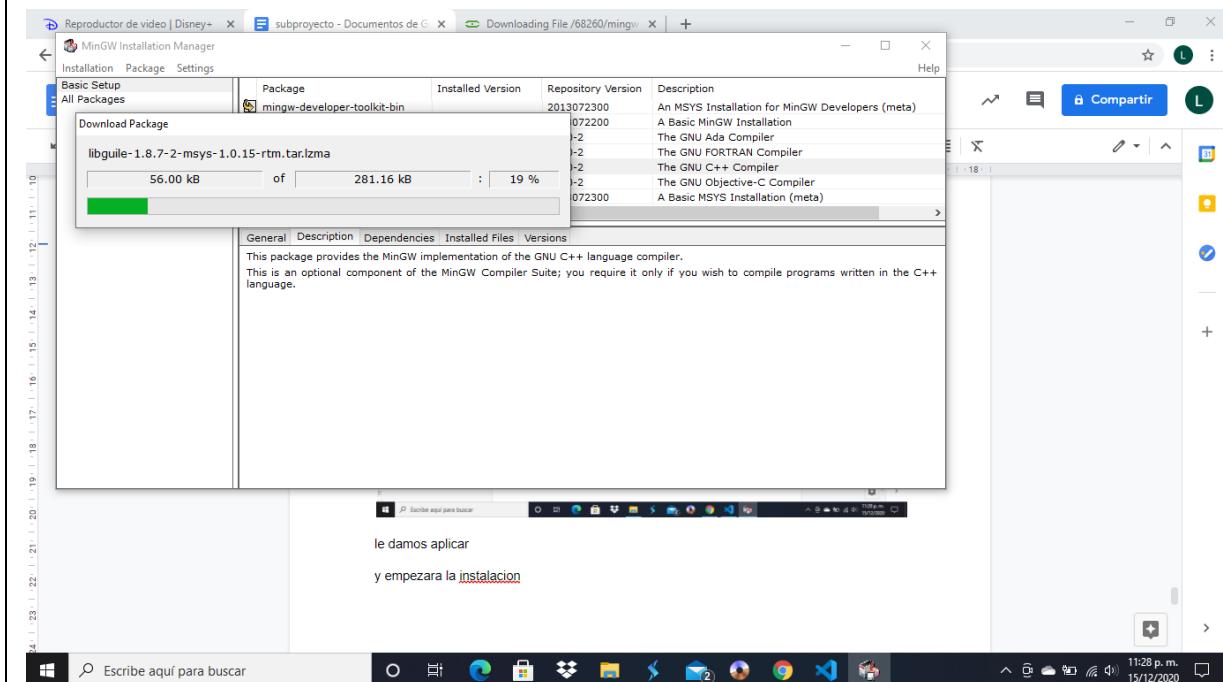
Escribe aquí para buscar

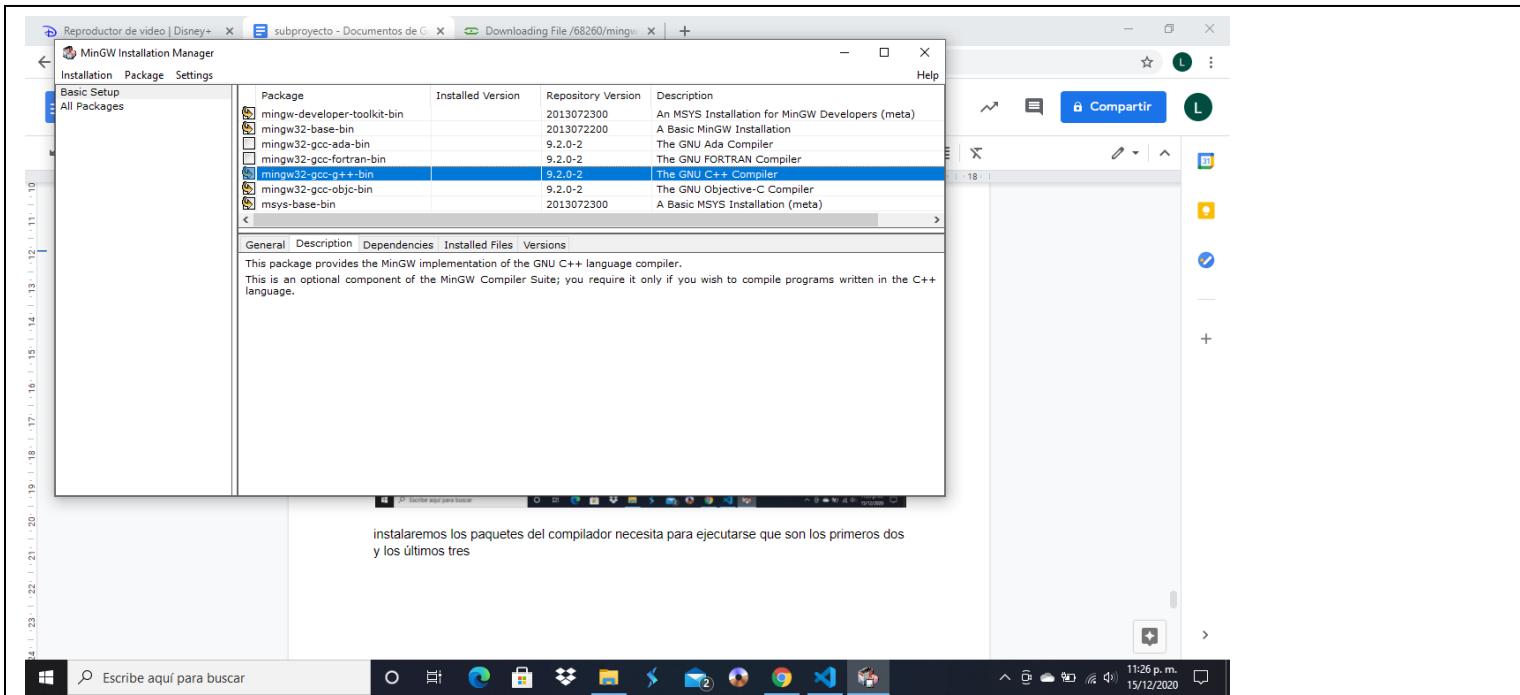
11:21 p.m. 15/12/2020

ya que terminamos de instalar le damos a continuar

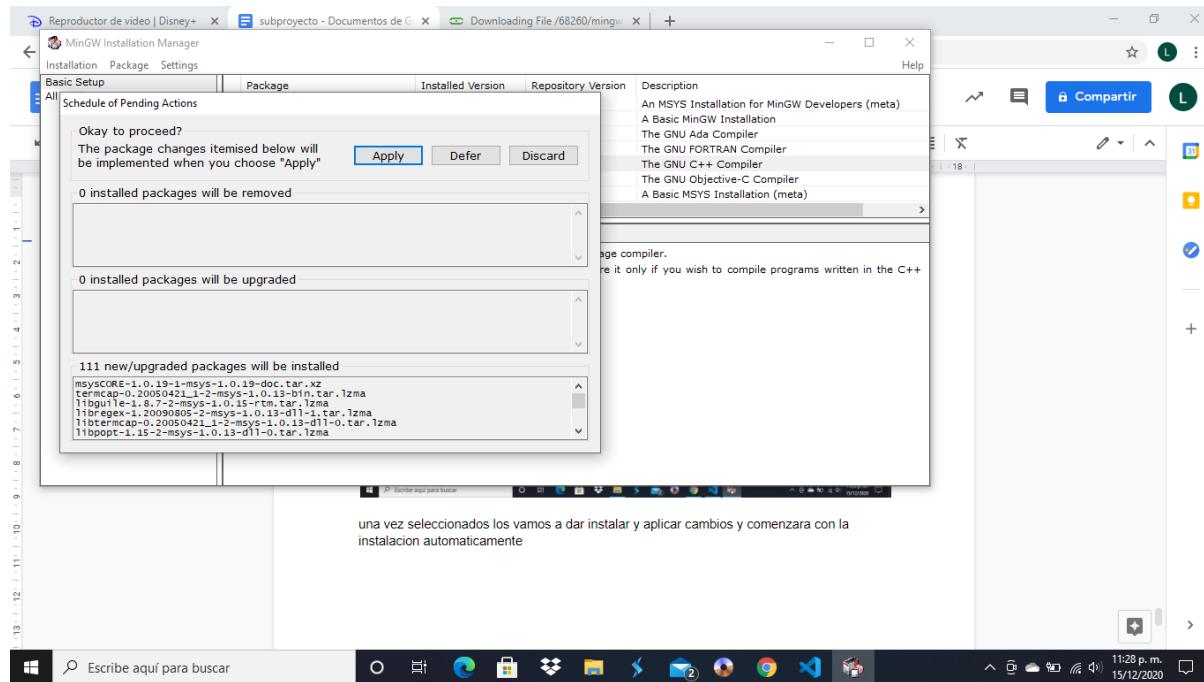


instalaremos los paquetes del compilador necesaria para ejecutarse que son los primeros dos y los últimos tres



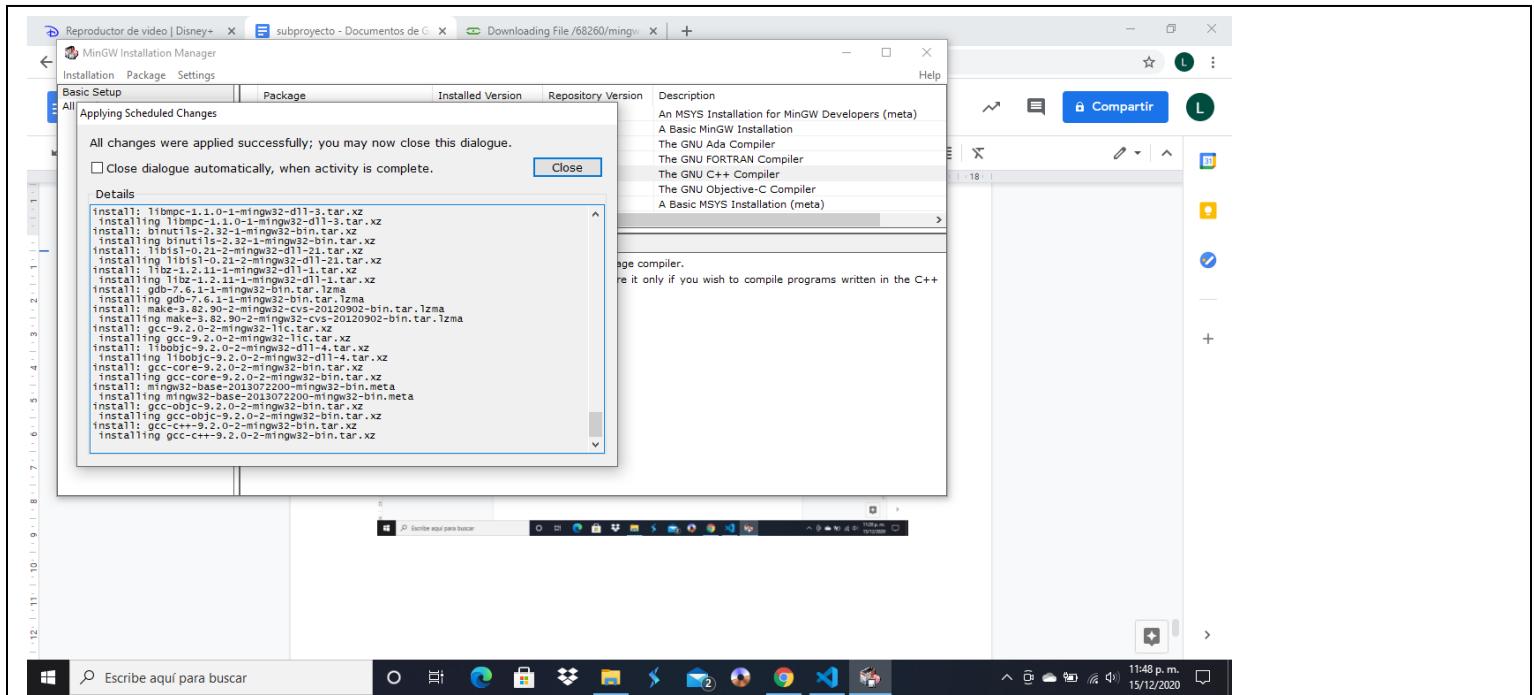


una vez seleccionados los vamos a dar instalar y aplicar cambios y comenzara con la instalación automáticamente

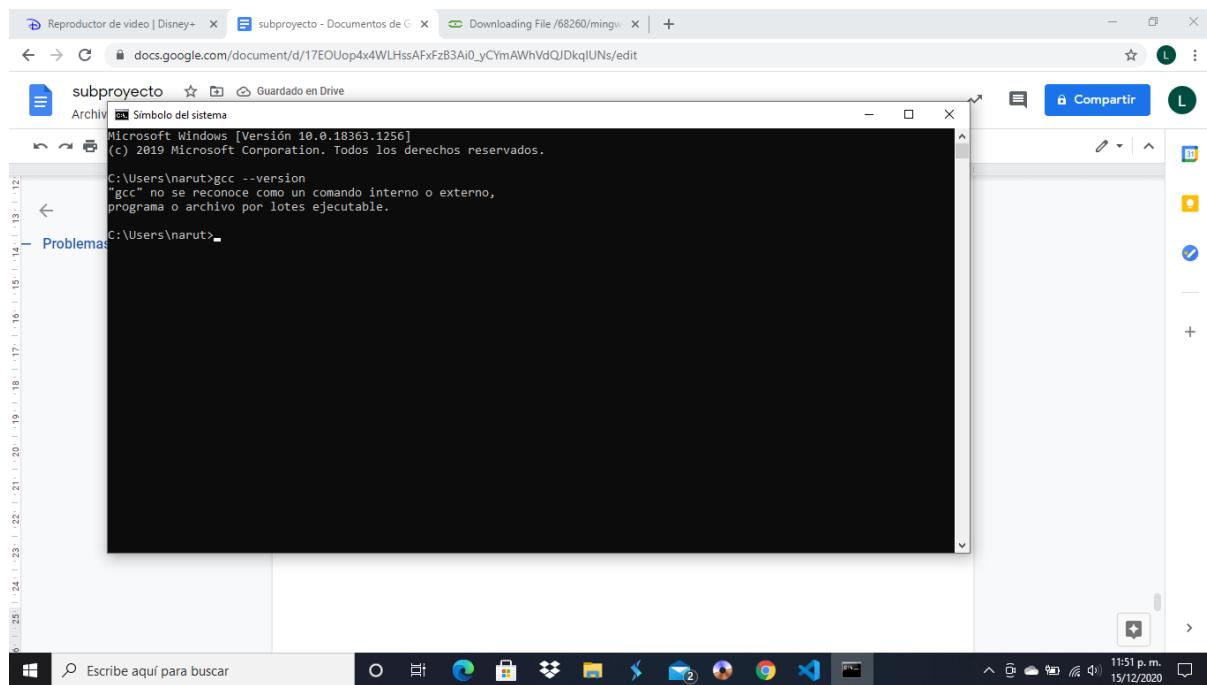


le damos aplicar

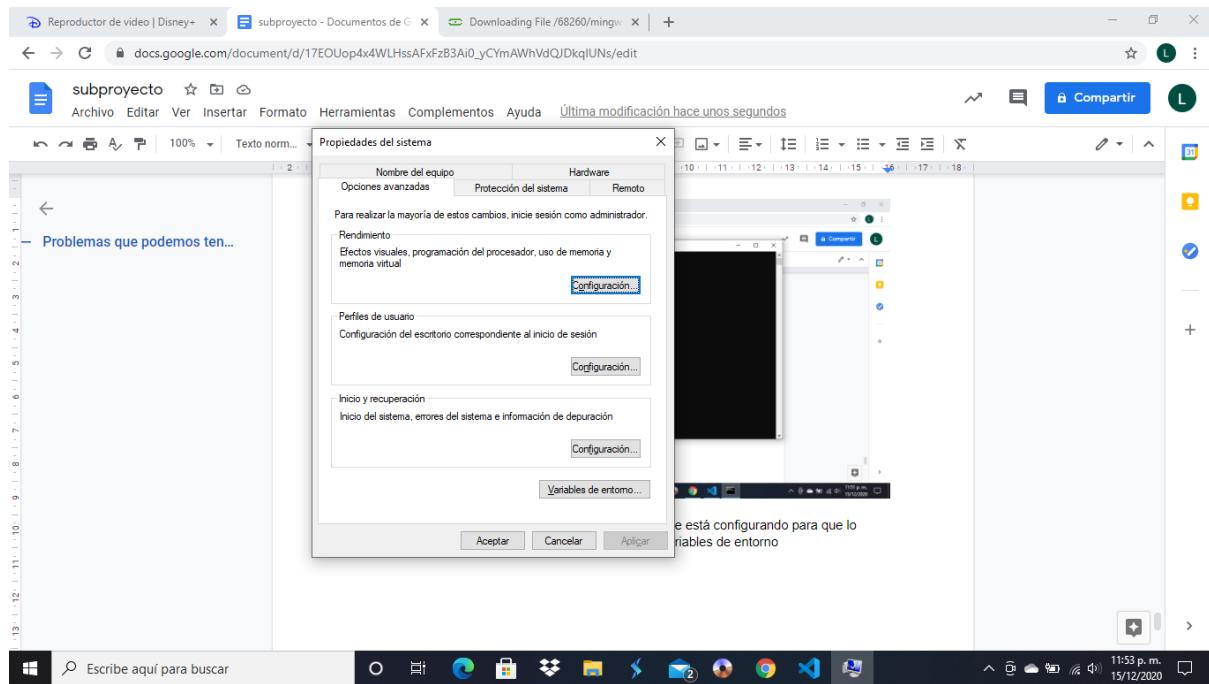
y empezará la instalación al terminarla instalación cerramos mingw



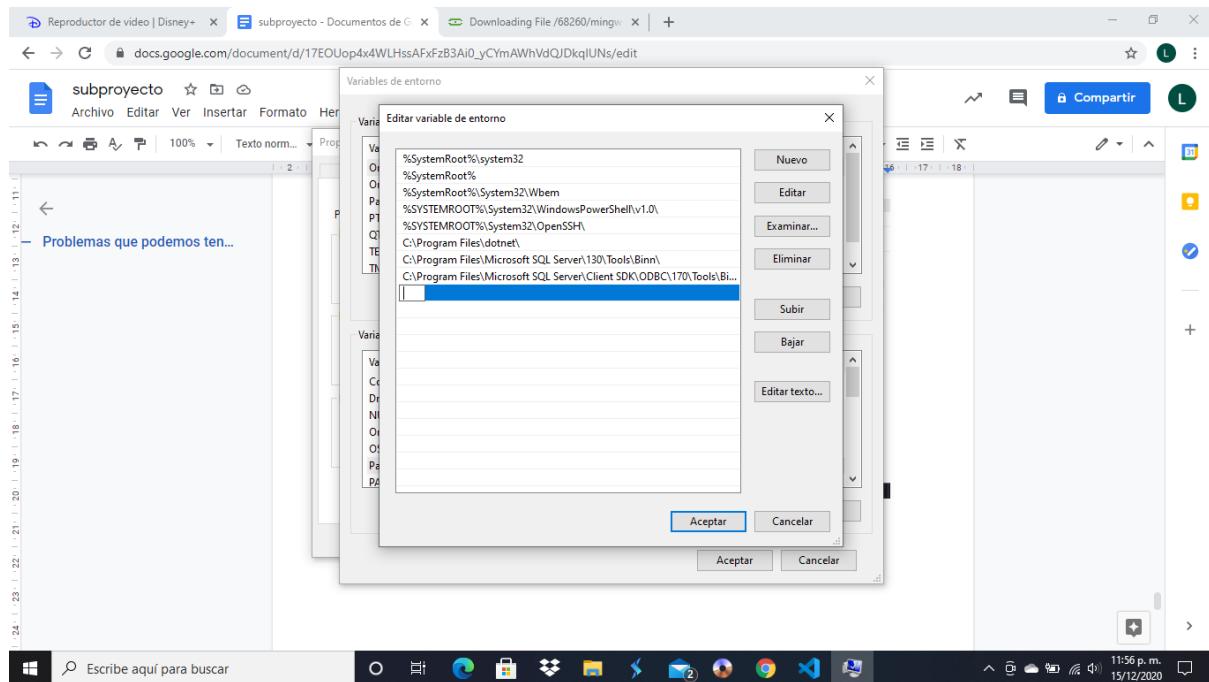
vamos a comprobar que el sistema operativo que reconoce nuestro compilador nos vamos a cmd y vamos a poner el código `gcc --version`



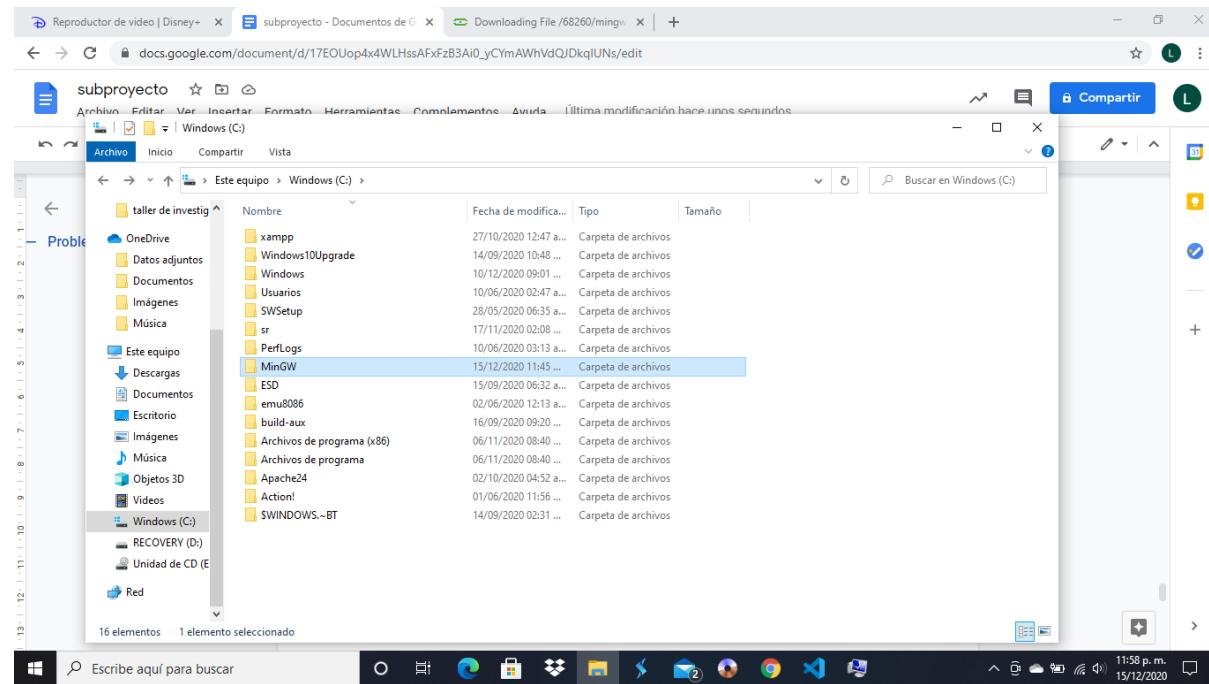
ya que el sistema operativo no lo reconoce tenemos que está configurando para que lo reconozca cerraremos cmd y nos vamos a editar las variables de entorno



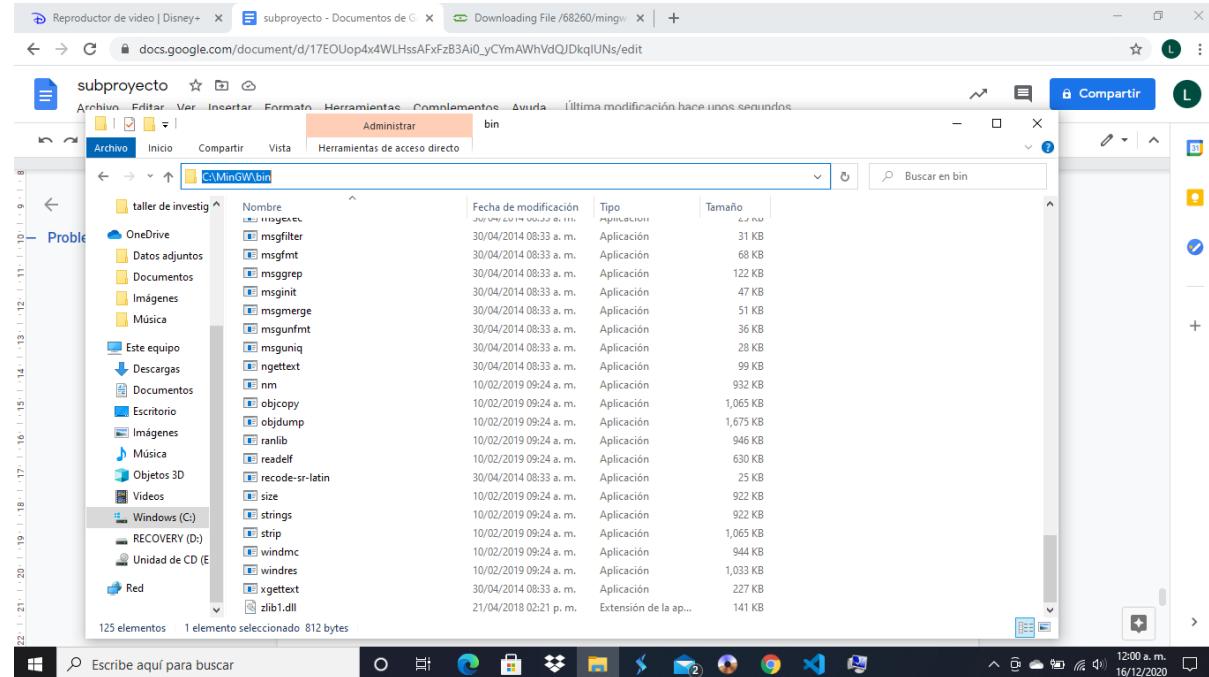
ya que estamos dentro nos vamos a donde dice variables de entorno le damos click luego buscaremos de abajo donde dice path y le damos editar



aquí es donde vamos a agregar la ruta donde se instaló nuestro programa nos vamos a nuestro disco local c y buscamos nuestro compilador MINGW y le damos click



esta carpeta abrimos y copiamos el enlace



y nos vamos a nuestro editor de variables de entorno y pegamos nuestro enlace de mingw y le damos aceptar y automáticamente se cierra y ahora vamos a comprobar que nuestro sistema operativo reconoce nuestro compilador



Reproductor de video | Disney+ | subprojeto - Documentos de Google | Downloading File /68260/mingw | +

Microsoft Windows [Versión 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

```
C:\Users\narut>gcc --version
gcc (MinGW.org GCC Build-2) 9.2.0
Copyright (C) 2019 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:\Users\narut>
```

121 elementos | 1 elemento seleccionado (101 bytes)

17 | 18

Compartir L

12:05 a. m. 16/12/2020

y nos vamos a nuestro editor de variables de entorno y pegamos nuestro enlace de mingw y le damos aceptar y automáticamente se cierra y ahora vamos a comprobar que nuestro sistema operativo reconoce nuestro compilador

nos vamos a cmd y ingresamos el comando

Escribe aquí para buscar

12:05 a. m. 16/12/2020

nos vamos a cmd y ingresamos el comando y vemos como nuestro sistema operativo reconoce nuestro compilador abrimos nuestro visual studio code y instalamos nuestro c\c++ y tambien c\c++ compile run

File Edit Selection View Go Run Terminal Help

Extension: C/C++ - automatas 2 - Visual Studio Code

EXTENSIONS: MA... 🔍 ⚙️ ...

C/C++ 1.13

C/C++ IntelliSense, debugging, and code browsing. Microsoft

Kite AI Code AutoComplete... 0.135.0

Tabnine Autocomplete AI: Ja... 3.1.18

C/C++ Compile Run 1.0.13

C/C++ Clang Command Adapt... 0.2.3

C/C++ Snippets 0.0.15

C# 1.23.7

Code Runner 0.11.1

Debugger for Chrome 4.12.11

C/C++ GNU Global 0.3.3

Extension: C/C++ x

C/C++ ms-vscode.cpptools Microsoft 16.037.850 ★★★★☆ Repository License v1.1.3

C/C++ IntelliSense, debugging, and code browsing.

Disable Uninstall This extension is enabled globally.

This extension is recommended based on the files you recently opened.

Details Feature Contributions Changelog

C/C++ for Visual Studio Code

Repository Issues Documentation Code Samples Offline Installers

Live Share enabled

The C/C++ extension adds language support for C/C++ to Visual Studio Code, including features such as IntelliSense and debugging.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

> Executing task: C/C++: cl.exe build active file <

Starting build...

Build finished with errors(s):

"cl.exe" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

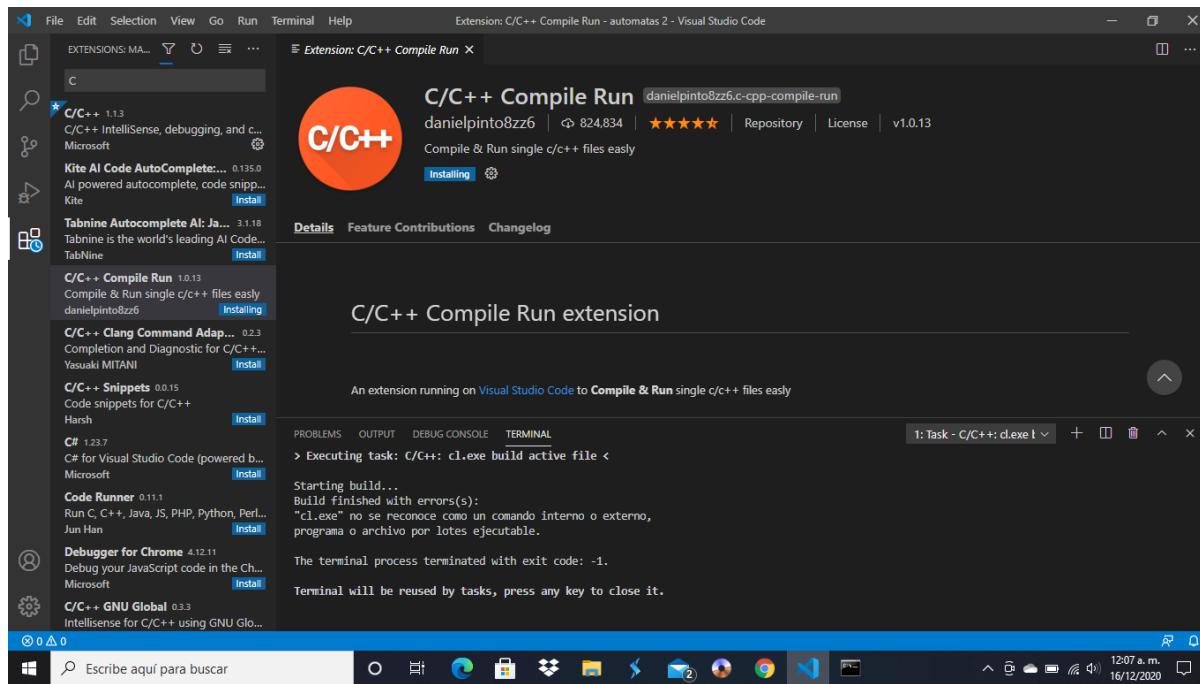
The terminal process terminated with exit code: -1.

Terminal will be reused by tasks, press any key to close it.

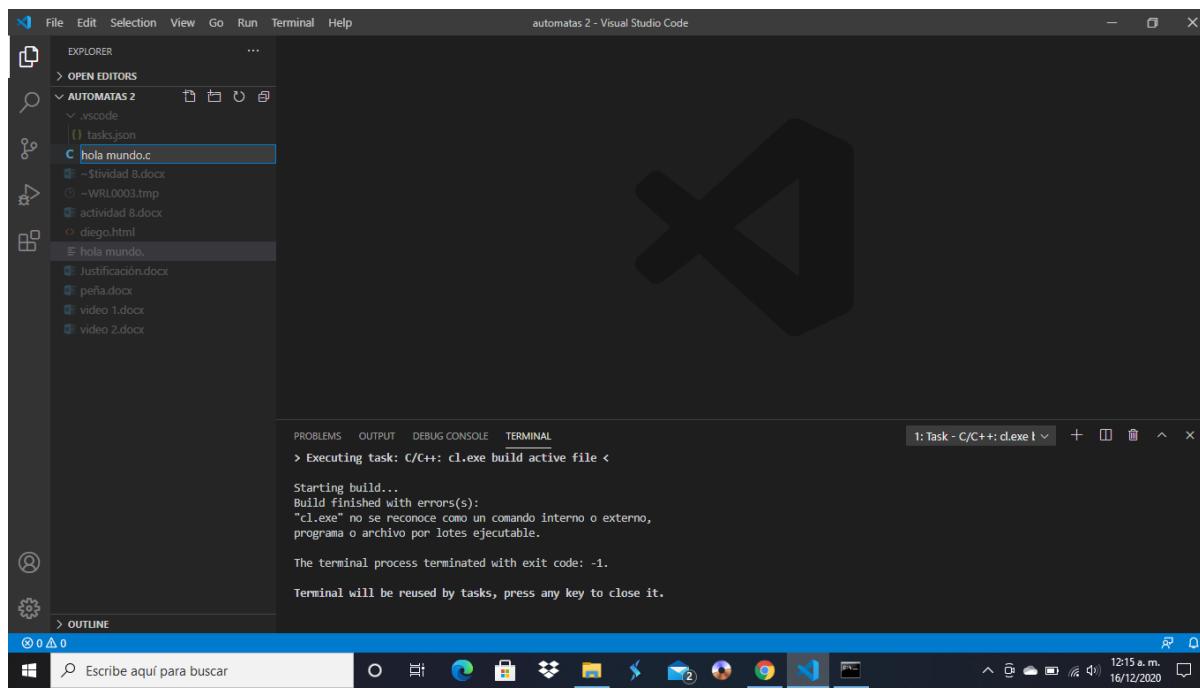
1: Task - C/C++: cl.exe

Escribe aquí para buscar

12:07 a. m. 16/12/2020



vamos a crear una carpeta en ese caso yo estoy utilizando la carpeta autómatas 2 donde guardo los programas agregamos un nuevo archivo que va ser hola mundo que va ser nuestro primer programa algo muy importante de agregar la extensión .c para que nuestro visual code que reconozca que estamos programando en c



The screenshot shows the Visual Studio Code interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the file is "holamundo.c - automatas 2 - Visual Studio Code". The Explorer sidebar on the left lists open editors ("OPEN EDITORS 1 UNSAVED") and files in ".vscode" and "AUTOMATAS 2" folders. The main editor area displays the following C code:

```
#include <stdio.h>
int main()
{
    printf("holamundo");
    return 0;
}
```

The status bar at the bottom shows Ln 6, Col 14, Spaces: 4, UTF-8, C, Win32, and system icons for battery, signal, and date/time.

le damos enter y nos crea nuestro archivo y empezamos a escribir nuestro programa

listo ya que tenemos nuestro programa vamos a guardar nuestro programa y compilamos y nos aparece esto le damos en la primera

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists files under "AUTOMATAS 2" and "holamundo.c".
- Code Editor:** The main area contains the following C code:

```
int main()
{
    printf("hola mundo");
    return 0;
}
```
- Environment Selection:** A dropdown menu titled "Select Environment" is open, showing "C++ (GDB/LLDB)" as the selected option, along with "C++ (Windows)" and "More...".
- Status Bar:** At the bottom, it displays "holamundo.c - automatas 2 - Visual Studio Code", "Ln 6 Col 14", "Spaces: 4", "UTF-8", "CRLF", "12:21 a.m.", "16/12/2020", and icons for search, file operations, and terminal.

y le damos al compilador que hemos descargado



```
version: "0.2.0",
"configurations": [
  {
    "name": "gcc.exe - Compilar y depurar el archivo activo",
    "type": "cppdbg",
    "request": "launch",
    "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",
    "args": [],
    "stopAtEntry": false,
    "cwd": "C:\\MinGW\\bin",
    "environment": [],
    "externalConsole": false,
    "MIMode": "gdb",
    "miDebuggerPath": "C:\\MinGW\\bin\\gdb.exe",
    "miDebuggerPort": 12345
  }
]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

> Executing task: C/C++: gcc.exe build active file

Starting build...

Build finished with errors(s):

c:/mingw/bin../lib/gcc/mingw32/9.2.0/../../../../mingw32/bin/id.exe: c:/mingw/bin../lib/gcc/mingw32/9.2.0/../../../../libmingw32.a(main.o):(.text.s+0x0): undefined reference to '_WinMain@16'

collect2.exe: error: ld returned 1 exit status

The terminal process terminated with exit code: -1.

Terminal will be reused by tasks, press any key to close it.

1: C++: gcc

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JSON with Comments Win32

Escribe aquí para buscar

y automáticamente nos manda a esta ventana solo vamos a modificar donde dice “externalconsole”: false a true que es para activarlo y le damos a guardar y lo corremos

```
version: "0.2.0",
"configurations": [
  {
    "name": "gcc.exe - Compilar y depurar el archivo activo",
    "type": "cppdbg",
    "request": "launch",
    "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",
    "args": [],
    "stopAtEntry": false,
    "cwd": "C:\\MinGW\\bin",
    "environment": [],
    "externalConsole": true,
    "MIMode": "gdb",
    "miDebuggerPath": "C:\\MinGW\\bin\\gdb.exe",
    "miDebuggerPort": 12345
  }
]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Warning: Debugger TargetArchitecture not detected, assuming x86_64.

=cmd-param-changed,param="pagination",value="off"

[New Thread 7400.0x7ec]

[New Thread 7400.0x2bd8]

Loaded 'C:\\WINDOWS\\System32\\kernel32.dll'. Symbols loaded.

Loaded 'C:\\WINDOWS\\System32\\KernelBase.dll'. Symbols loaded.

Loaded 'C:\\WINDOWS\\System32\\apphelp.dll'. Symbols loaded.

Loaded 'C:\\WINDOWS\\System32\\msvcrt.dll'. Symbols loaded.

The program 'c:\\Users\\harut\\Desktop\\automatas 2\\hola mundo.exe' has exited with code 0 (0x00000000).

Ln 3, Col 7 Spaces: 4 UTF-8 CRLF C Win32

Escribe aquí para buscar



```
#include <stdio.h>
int main()
{
    printf("holamundo");
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Warning: Debuggee TargetArchitecture not detected, assuming x86_64.
-cmd-param-changed,param="pagination",value="off"
[New Thread 7400.0x7ec]
[New Thread 7400.0xb2d0]
Loaded 'C:\WINDOWS\SysWOW64\kernel32.dll'. Symbols loaded.
Loaded 'C:\WINDOWS\SysWOW64\KernelBase.dll'. Symbols loaded.
Loaded 'C:\WINDOWS\SysWOW64\apphelp.dll'. Symbols loaded.
Loaded 'C:\WINDOWS\SysWOW64\msvcrtd.dll'. Symbols loaded.
The program 'c:\Users\narut\Desktop\automatas 2\holamundo.exe' has exited with code 0 (0x00000000).

Escribe aquí para buscar

nos vamos al explore donde a creado un archivo .exe

```
#include <stdio.h>
int main()
{
    printf("holamundo");
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Warning: Debuggee TargetArchitecture not detected, assuming x86_64.
-cmd-param-changed,param="pagination",value="off"
[New Thread 7400.0x7ec]
[New Thread 7400.0xb2d0]
Loaded 'C:\WINDOWS\SysWOW64\kernel32.dll'. Symbols loaded.
Loaded 'C:\WINDOWS\SysWOW64\KernelBase.dll'. Symbols loaded.
Loaded 'C:\WINDOWS\SysWOW64\apphelp.dll'. Symbols loaded.
Loaded 'C:\WINDOWS\SysWOW64\msvcrtd.dll'. Symbols loaded.
The program 'c:\Users\narut\Desktop\automatas 2\holamundo.exe' has exited with code 0 (0x00000000).

Escribe aquí para buscar

y nos copila nos dice hola mundo



holamundo.c - automatas 2 - Visual Studio Code

```
1 #include <stdio.h>
```

Terminal will be reused by tasks, press any key to close it.

gcc.exe - Compilar y depurar el archivo activo (automatas 2)

Escribe aquí para buscar

File Edit Selection View Go Run Terminal Help

EXPLORER OPEN EDITORS AUTOMATAS 2 .vscode

launch.json tasks.json hola mundoPresione una tecla para continuar . . .

~\$ividad ~WRL000 actividad diego.htm hola mun hola mun Justificaci peña.doc video 1.d video 2.dk

OUTLINE

Ln 6 Col 21 Spaces: 4 UTF-8 CRLF C Win32 12:51 a.m. 16/12/2020

Instalación de msys2 y clang

Aquí vamos a descarga el msys2 para que podamos trabajar para nuestro proyecto de clang y se va descargar automáticamente

MSYS2 download | SourceForge

sourceforge.net/projects/msys2/

SOURCEFORGE

Open Source Software Business Software Resources Help Create Join Login

Search for software or solutions

MSYS2

Software Distribution and Building Platform for Windows

Brought to you by: ellieux, lexx83, mingwandroid, nixmann

★★★★★ 40 Reviews Downloads: 1,207,057 This Week Last Update: 5 days ago

Download Get Updates Share This

Summary Files Reviews Support Wiki Tickets Discussion Mailing Lists External Link

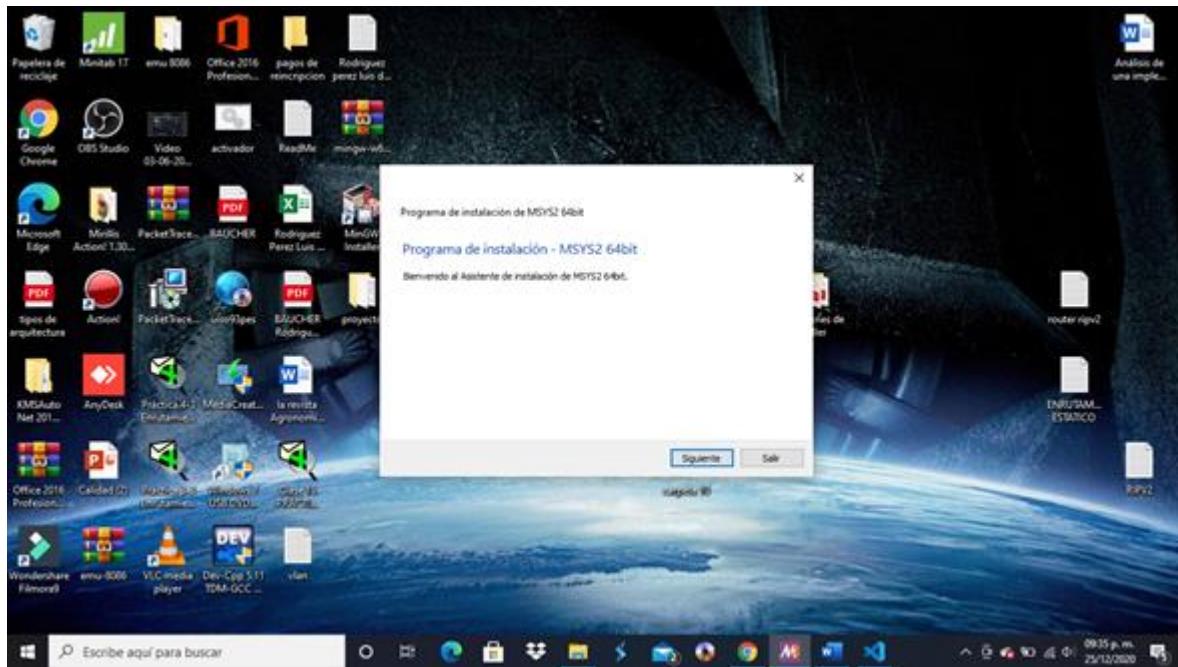
MSYS2 is a collection of tools and libraries providing you with an easy-to-use environment for building, installing and running native Windows software.

It consists of a command line terminal called minTTY, bash, version control systems like git and subversion, tools like tar and awk and even build

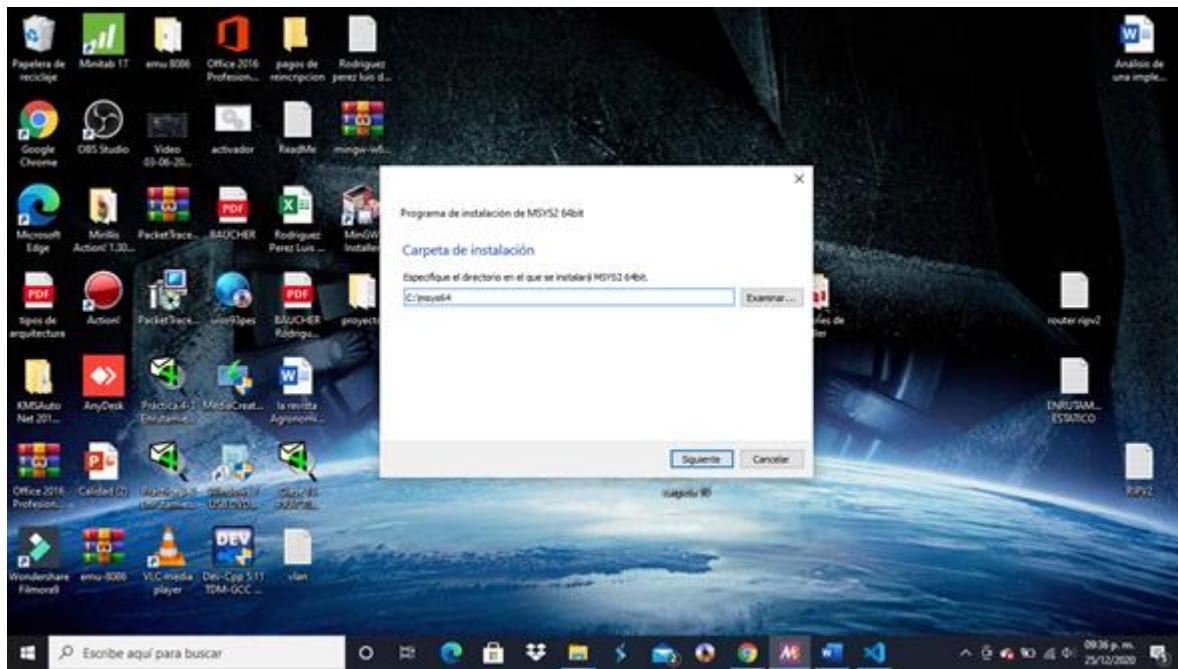
Escribe aquí para buscar



Y empecemos con la instalación de Msys2

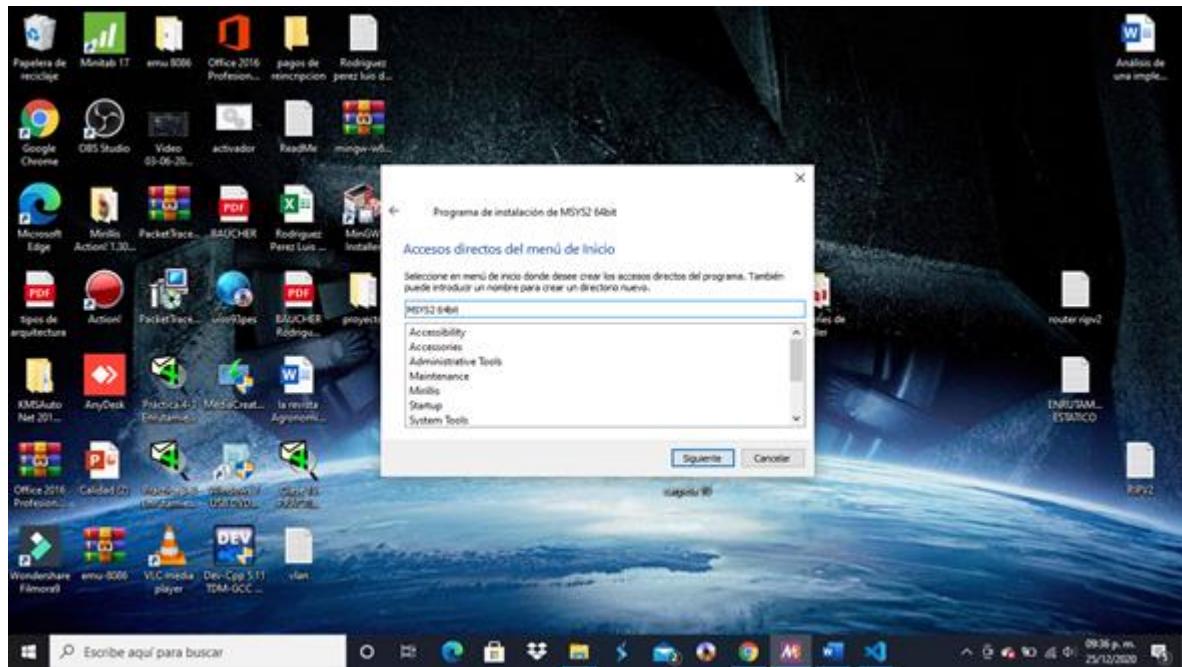


Aquí nos manda donde vamos a tener Msys2 que va ser en nuestro disco local para que este instalado nuestro Msys2

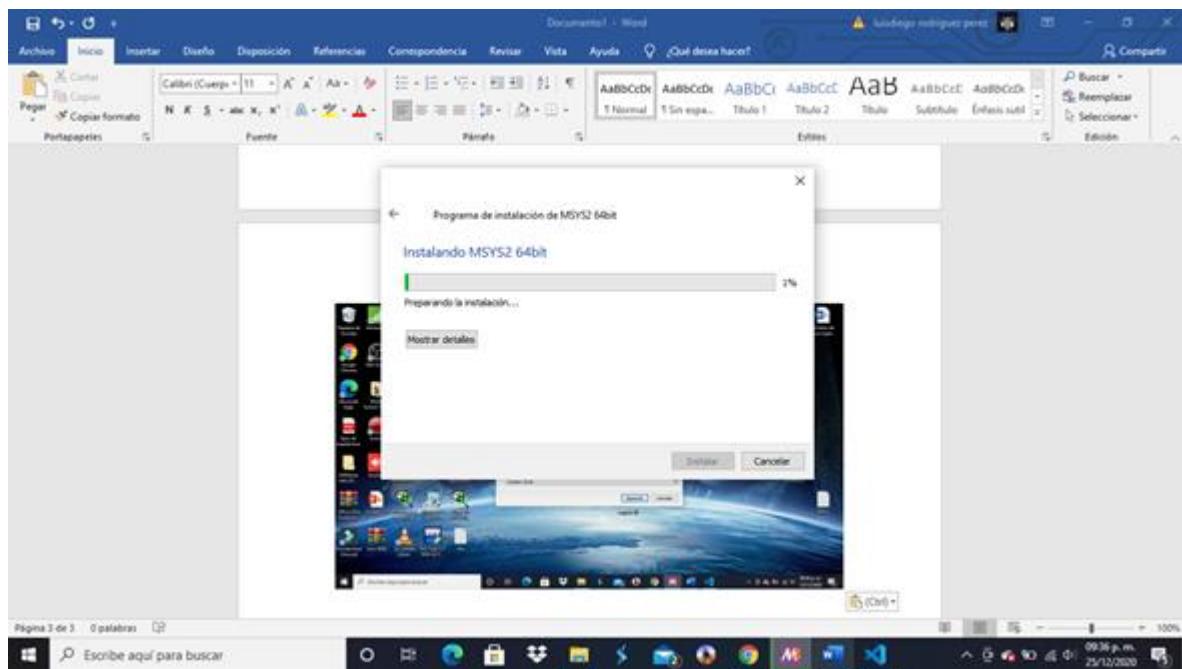




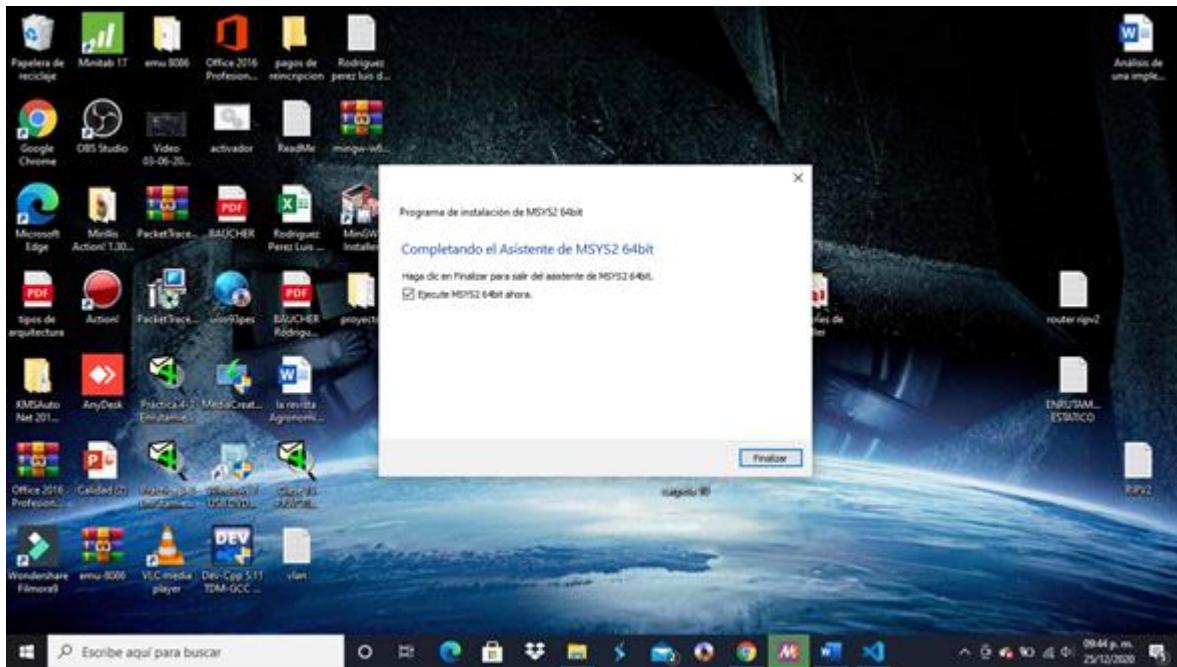
Aquí vemos lo que va instalar los accesos directos del menú de inicio le damos siguiente



Y se va instalando nuestro Msys2



Y ya está instalando el Msys2 le damos finalizar



Ahora abrimos Msys y ponemos el comando pacman -Syu donde nos va decir si queremos instalar los paquetes que se van a utilizar

```
M - 
DESKTOP-LP0GK940VG MSYS - 
$ pacman -Sya
    Sincronizando las bases de datos de los paquetes...
microsoft-xml-3.0-1 787.8 KiB 326 KiB/s 00:02 [####] 100%
microsoft-xml-3.0-1.sig 438.0 B 0.00 B/s 00:00 [####] 100%
microsoft-xml-3.0-1.r 789.7 KiB 322 KiB/s 00:02 [####] 100%
microsoft-xml-3.0-1.r.sig 438.0 B 0.00 B/s 00:00 [####] 100%
microsoft-xml-3.0-1.m 288.6 KiB 324 KiB/s 00:01 [####] 100%
microsoft-xml-3.0-1.m.sig 438.0 B 0.00 B/s 00:00 [####] 100%
  Verificando core system upgrade...
advertencia: terminar otras MSYS2 programs before proceeding
resolviendo dependencias...
buscando conflictos entre paquetes...

Paquetes (5) bash-5.1.004-1 minTTY-1-3.4.4-1 msys2-runtime-3.1.7-4
  pacman-5.2.2-7 pacman-mirrors-20201208-1

Tamaño total de la descarga: 11.02 MiB
Tamaño total de la instalación: 53.77 MiB
Tamaño neto tras actualizar: <1.24 MiB

:: ¿Continuar con la instalación? [S/n] |
```

Y seguimos instalando paquetes ya cuando tengamos instalando los paquetes ahora vamos a ver qué más tenemos que instalar

```
pacman -Syu
:: Sincronizando las bases de datos de los paquetes...
mingw32-base 5.1.0-1 32bit 1.7M 0% [=====] 100%
mingw32.sig 438.0 B 0.00 B/s 00:00 [=====] 100%
mingw44-base 5.1.0-1 32bit 1.7M 0% [=====] 100%
mingw44.sig 438.0 B 0.00 B/s 00:00 [=====] 100%
sysv 288.6 Kib 324 KiB/s 00:01 [=====] 100%
sysv.sig 438.0 B 0.00 B/s 00:00 [=====] 100%
:: Starting core system upgrade...
advertisment: terminando other MSYS2 programs before proceeding
resolvingo dependencias...
buscando conflictos entre paquetes...
Paquetes (5) bash-5.1.004-1 mingw32-1-3.6.4-1 sysv-20200120-1
pacman-5.2.2-7 pacman-mirrors-20200120-1

Tamaño total de la descarga: 31.02 MiB
Tamaño total de la instalación: 33.77 MiB
Tamaño neto tras actualizar: -2.74 MiB

:: ¿Continuar con la instalación? [S/n] S

:: Obteniendo los paquetes...
bash-5.1.004-1-x86_64 2.3 MiB 337 KiB/s 00:07 [=====] 100%
mingw32-1-3.6.4-1-x86_64 787.2 KiB 229 KiB/s 00:03 [=====] 100%
sysv-20200120-1 2.6 MiB 128 KiB/s 00:21 [=====] 100%
pacman-5.2.2-7-x86_64 5.4 MiB 359 KiB/s 00:15 [=====] 100%
pacman-mirrors-20200120-1-x86_64 0 B 0.00 B/s 00:00 [=====] 100%

(5/5) comprobando las claves del depósito
(5/5) verificando la integridad de los paquetes
(5/5) cargando los archivos de los paquetes
(5/5) comprobando conflictos entre archivos
(5/5) comprobando el espacio disponible en el disco
(5/5) actualizando los índices de los paquetes...
(5/5) actualizando bash
(5/5) actualizando mingw32
(5/5) actualizando sysv-runtime
(5/5) actualizando pacman-mirrors
(5/5) actualizando pacman
:: Ejecutando los hooks de postinstalación...
(5/5) ejecutando /etc/pacman.d/hooks/postinst

6 [error] pacman(4463): C:\cygwin\usr\bin\pacman.exe: *** fatal error - cygheap base mismatch detected - 0x180348408/0x180345408.
This problem is probably due to using incompatible versions of the cygwin DLL.
Search for cygwin.dll using the Windows Start+Find/Search facility
and delete all but the most recent version. The most recent version "should"
reside in c:\cygwin\bin, where 'c:' is the drive on which you have
installed the cygwin distribution. Rebooting is also suggested if you
are unable to find another cygwin.dll.

0 [error] pacman(4463): child 12372 died unexpectedly, retry 0, exit code 0xC0000042, errno 11
error: no se puede crear un nuevo proceso (Resumen temporalmente unavailable)
:: To complete this update all MSYS2 processes including this terminal will be closed. Confirm to proceed [S/n] |
```

Ahora seguimos más instalación de paquetes con el comando pacman -Syu

```
Administrator:~ pacman -Syu
:: Synchronizing package databases...
mingw32 is up to date
mingw64 is up to date
msys is up to date
:: Starting core system upgrade...
there is nothing to do
:: Starting optional system upgrade...
resolving dependencies...
looking for conflicting packages...

Packages (20) base-2020.12-1  bsdtar-3.5.0-1  curl-7.74.0-4  glibc-2.26.4-3
               grep-6.2.1-1  gnupg-2.2.25-1  libassuan-2.5.4-1  libcurl-7.74.0-4
               libedit-20191231.3.1-2  libgnutls-3.7.0-1  libgpgme-1.25.0-1
               libidn2-2.3.0  liblber-2.6.0-1  liblber-2.6.0-1  liblber-2.6.0-1
               libreadline-8.1.0-1  libsigntools-9.34.0-1  libstdc++-4.8-3
               nano-5.4-3  openssl-1.1.1-i-1  zstd-1.4.6-2

Total Download Size: 32.82 MiB
Total Installed Size: 44.25 MiB
Net Upgrade Size: 3.01 MiB

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



```
[#] - # X
libbz2=2.5.4-1+b6,64
libgdiplus=3.7.0-1+b6,64
gnome2=2.2.25-5+squeeze,64
Tlbgpcre=1.15.0-0.1+b6,64
Tlbgpcre2=8.10.36-1+b6,64
base=2020.12-1-any
(20/20) checking keys in keyring
(20/20) checking package integrity
(20/20) checking package files
(20/20) checking for file conflicts
(20/20) checking available disk space
:: Processing package changes...
( 1/20) upgrading Tlbgz
( 2/20) upgrading Tlbgzstd
( 3/20) upgrading gpp
( 4/20) upgrading libreadline
( 5/20) upgrading libtar
( 6/20) upgrading libxml2
( 7/20) upgrading openssl
( 8/20) upgrading glibz
( 9/20) upgrading Tlbgsqlite
(10/20) upgrading libedit
(11/20) upgrading libcurl
(12/20) upgrading curl
(13/20) installing nano
(14/20) upgrading curl
(15/20) upgrading Tlbgssuan
(16/20) upgrading Tlbgmutils
(17/20) upgrading gnugc
=> Appending keys from msys2.gpg...
=> Locally signing trusted keys in keyring...
-> Locally signing key 12304051A179859C208E916888E514E53E0000813...
-> Locally signing key B91BCF330329590CC0E916888E514E53E000081900...
-> Locally signing key 4054E94D940790AE56A1E824C021878...
-> Locally signing key 69985C5E851011C7A0E75A075A8182AC0218578...
-> Locally signing key 900004217071A13B8896159E8047EF24BAEEA755C...
-> Locally signing key 055E7A407CE98A1587C0ACFC400263ECA25678A...
=> Importing owner trust values
=> Disabling revoked keys in keyring...
-> Disabling key B1914FB55E836684718296E7940CF9799FC717...
=> Updating trust database...
=> Next trust check due at 2021-10-02
(18/20) upgrading Tlbgpcre
(19/20) upgrading Tlbgpcre2_8
(20/20) upgrading base
:: Running post-transaction hooks...
(1/1) Updating the info directory file...

KEYBOARDLAPTOP-0K:~$XWSV =
```

Ahora que terminamos de instalar los paquetes de Msys2 ahora vamos a instalar clang para que podamos programar con clang con el comando pacman -S mingw-x64-x86_64-clang ya nos manda los posibles clang que hay en Msys2

Y instalamos los paquetes de clang para poder programar

Aquí vamos a instalar nano a que me refiero nano digamos es donde vamos a poder programar digamos que va ser nuestro lenguaje y instalamos los paquetes que se va utilizar

```
[...]
[DESKTOP-GOK96QV: ~] pacman -S nano vim
warning: nano-5.4-1 is up to date -- reinstalling
error: target not found: vim

[DESKTOP-GOK96QV: ~] pacman -S nano vim
warning: nano-5.4-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

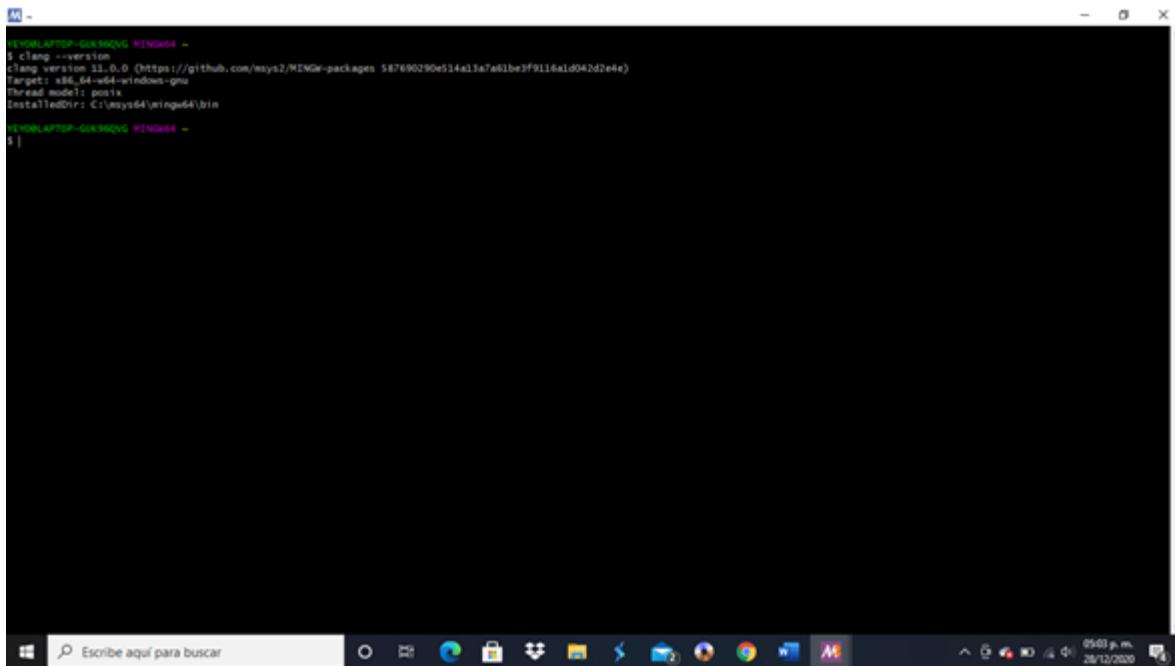
Packages (2) nano-5.4-1 vim-8.2.1891-1

Total Download Size: 7.78 MiB
Total Installed Size: 50.28 MiB
Net Upgrade Size: 47.67 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
:: vime-8.2.1891-1-x86_64
(2/2) checking keys in keyring
(2/2) checking package integrity
(2/2) loading package files
(2/2) checking for file conflicts
(2/2) checking available disk space
(2/2) processing package changes...
(2/2) installing vim
(2/2) installing vim
:: Running post-transaction hooks...
(1/1) Updating the info directory file...

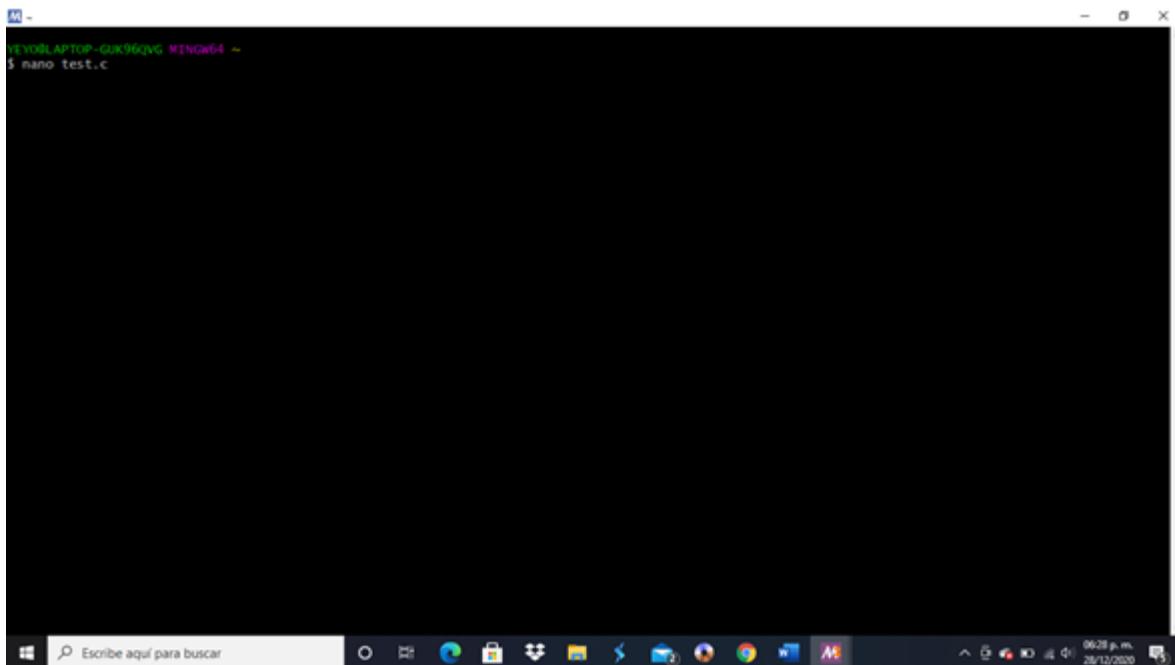
[DESKTOP-GOK96QV: ~] s |
```

Ya que terminamos la instalación de nano ahora vemos que versión de clang tenemos con el comando clang –versión y automáticamente nos manda que versión tenemos



```
KEVOLAPTOP-GEN96QVG MINGW64 ~
$ clang --version
clang version 11.0.0 (https://github.com/msys2/MINGW-packages 587690290e514a13a7a61be3f9116a1d042d2e4e)
Target: x86_64-w64-windows-gnu
Thread model: posix
InstalledDir: C:/msys64/mingw64/bin
KEVOLAPTOP-GEN96QVG MINGW64 ~
$ |
```

Ahora aquí vamos a empezar a programar ponemos el comando nano donde vamos a programar test.c es el nombre de nuestro programa le damos enter



```
KEVOLAPTOP-GEN96QVG MINGW64 ~
$ nano test.c
```

Aquí es donde vamos a programar que va ser en c donde vamos a imprimir el hello from clang como muestra en la imagen ya que terminamos lo vamos a guardar con control + x nos va decir que si no o cancelar y le damos si y va guardarse



GNU nano 5.4
#include <stdio.h>
int main (void){
 printf("Hello from clang\n");
}

The terminal window has a menu bar with Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, Copy, To Bracket, and Where Was. A status bar at the bottom shows "Escribe aquí para buscar", the date "28/12/2020", and the time "06:19 p.m."

Ya que terminamos de programar cerramos el nano y vamos a verificar que no tengamos un error ya si nos marca un error ahora si que nos marca que error podemos tener cualquiera como un punto coma o un punto etc.

Este es el comando

Clang -std=c17 -Wall -Wextra -pedantic test.c

Ya que no nos marca un error podemos este comando que es para correr nuestro programa donde vamos a ver nuestro mensaje que hicimos de nuestro programa

Con este comando

./a.exe

Y nos manda nuestro nuestro mensaje hello from clang

```
DEYORLALTOP-GUK96QVG MINGW64 ~
$ clang -std=c17 -Wall -Wextra -pedantic test.c
DEYORLALTOP-GUK96QVG MINGW64 ~
$ ./a.exe
Hello from clang
DEYORLALTOP-GUK96QVG MINGW64 ~
```

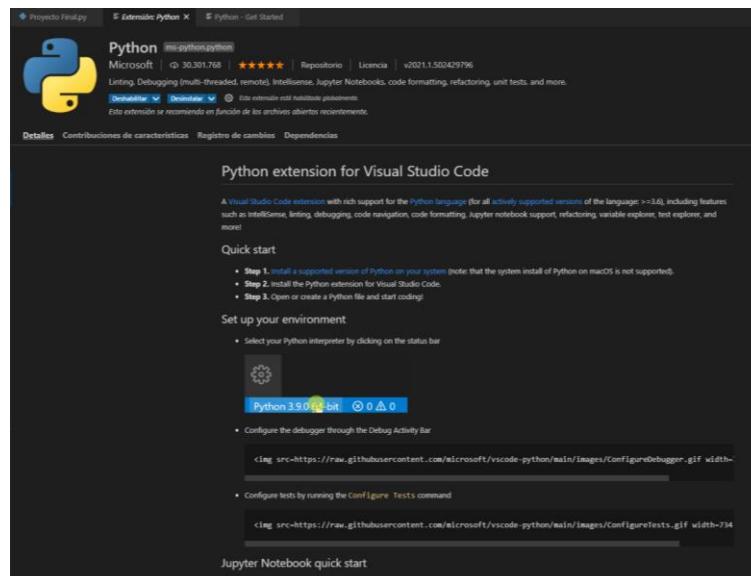
The terminal window has a menu bar with Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, Copy, To Bracket, and Where Was. A status bar at the bottom shows "Escribe aquí para buscar", the date "28/12/2020", and the time "06:20 p.m."

Instalación de Python

Se hace la descarga de Python en su página oficial y lo instalamos



Para realizar el Análisis Léxico, primero tenemos que instalar la librería de Python en Visual Studio Code:



Después tenemos que descargar las librerías locales: lex, yacc, etc. Y colocar su carpeta dónde vienen incluidas en la siguiente dirección:

C:\Users\feli_\AppData\Local\Programs\Python\Python39\Lib



o rápido			
	Nombre	Fecha de modificación	Tipo
	__pycache__	31/12/2020 08:02 p. m.	Carpeta de arch
	init.py	15/02/2018 12:44 p. m.	Python File
	cpp.py	15/02/2018 12:44 p. m.	Python File
	ctokens.py	15/02/2018 12:44 p. m.	Python File
	lex.py	15/02/2018 12:44 p. m.	Python File
	yacc.py	15/02/2018 12:44 p. m.	Python File
	ygen.py	15/02/2018 12:44 p. m.	Python File

Programa – Analizador Léxico:

En primera instancia, tenemos que importar la librería local “lex” para el funcionamiento del compilador.

```
import ply.lex as lex
```

Tenemos que declarar la variable donde se va a guardar el resultado del análisis teniendo en cuenta el lexema ingresado, en este caso el lexema es la cadena de caracteres o valor numérico aceptados por el token.

```
# resultado del análisis
resultado_lexema = []
```



En seguida tenemos que declarar cada uno de los tokens, empezamos por las palabras reservadas:

```
reservada = (
    # Palabras Reservadas
    'INCLUDE',
    'USING',
    'NAMESPACE',
    'STD',
    'COUT',
    'CIN',
    'GET',
    'CADENA',
    'RETURN',
    'VOID',
    'INT',
    'ENDL',
)
```

Seguimos creando los tokens, pero ahora con las palabras que definen las operaciones aritméticas, las condiciones, los ciclos, las compuertas lógicas, los símbolos (corchetes, llaves, etc.), y los signos de puntuación:

```
tokens = reservada + [
    'IDENTIFICADOR',
    'ENTERO',
    'ASIGNAR',

    'SUMA',
    'RESTA',
    'MULT',
    'DIV',
    'POTENCIA',
    'MODULO',
    'MINUSMINUS',
    'PLUSPLUS',
]

#Condiciones
'SI',
'SINO',
#Ciclos
'MIENTRAS',
'PARA',
#logica
'AND',
'OR',
'NOT',
'MENORQUE',
'MENORIGUAL',
'MAYORQUE',
'MAYORIGUAL',
'IGUAL',
'DISTINTO',
]

# Symbolos
'NUMERAL',
'PARIZQ',
'PARDER',
'CORIZQ',
'CORDER',
'LLAIZQ',
'LLADER',
]

# Otros
'PUNTOCOMA',
'COMA',
'COMDOB',
'MAYORDER', #>>
'MAYORIZQ', #<<
```

Posteriormente pasamos a la declaración de las Expresiones Regulares, definiremos los símbolos de aceptación que tomará cada token al momento de ingresar los lexemas para su análisis:

```
# Reglas de Expresiones Regulares para token de Contexto simple

t_SUMA = r'\+'
t_RESTA = r'-'
t_MINUSMINUS = r'\-\-'
t_PUNTO = r'\.'
t_MULT = r'\*'
t_DIV = r'/' 
t_MODULO = r'\%'
t_POTENCIA = r'(\*\{2\} | \^)'

t_ASIGNAR = r'='
# Expresiones Logicas
t_AND = r'\&\&'
t_OR = r'\|{2}'
t_NOT = r'\!'
t_MENORQUE = r'<'
t_MAYORQUE = r'>'
t_PUNTOCOMA = ';'
t_COMA = r'\,'
t_PARIZQ = r'\('
t_PARDER = r'\)'
t_CORIZQ = r'\['
t_CORDER = r'\]'
t_LLAIZQ = r'\{'
t_LLADER = r'\}'
t_CONDDOB = r'\\"
```

En seguida definiremos las funciones (def) necesarias de los tokens para reconocer los números, los identificadores, los símbolos, etc., tomando como referencia que son Expresiones Regulares avanzadas:



```
def t_INCLUDE(t):
    r'include'
    return t

def t_USING(t):
    r'using'
    return t

def t_NAMESPACE(t):
    r'namespace'
    return t

def t_STD(t):
    r'std'
    return t

def t_COUNT(t):
    r'cout'
    return t

def t_CIN(t):
    r'cin'
    return t

def t_GET(t):
    r'get'
    return t

def t_ENDL(t):
    r'endl'
    return t

def t_SI(t):
    r'if'
    return t

def t_RETURN(t):
    r'return'
    return t

def t_VOID(t):
    r'void'
    return t

def t_MIENTRAS(t):
    r'while'
    return t

def t_PARA(t):
    r'for'
    return t

def t_ENTERO(t):
    r'\d+'
    t.value = int(t.value)
    return t

def t_IDENTIFICADOR(t):
    r'\w+(\_\w\w)*'
    return t

def t_CADENA(t):
    r'\"?(\w+ \ * \w*\d* \ *)\"?'
    return t

def t_NUMERAL(t):
    r'\#'
    return t

def t_PLUSPLUS(t):
    r'\+\+'
    return t

def t_DISTINTO(t):
    r'!='
    return t

def t_MENORIGUAL(t):
    r'<='
    return t

def t_MAYORIGUAL(t):
    r'>='
    return t

def t_IGUAL(t):
    r'=='
    return t

def t_MAYORDER(t):
    r'<<'
    return t

def t_MAYORIZQ(t):
    r'>>'
    return t

def t_comments(t):
    r'/*(.|\n)*?\*/'
    t.lexer.lineno += t.value.count('\n')
    print("Comentario de multiple linea")

def t_comments_ONELine(t):
    r'//(.)*\n'
    t.lexer.lineno += 1
    print("Comentario de una linea")
    t_ignore = '\t'
```

Es de suma importancia agregar la función de error, esto es para que el análisis detecte errores cuando un carácter ingresado por el usuario no es válido:

```
def t_error( t ):
    global resultado_lexema
    estado = "*** Token no valido en la Linea {:4} Valor {:16} Posicion {:4}".format(str(t.lineno), str(t.value),
    resultado_lexema.append(estado)
    t.lexer.skip(1)
```



Casi para terminar, se agrega la función para hacer pruebas de ingreso de los lexemas. Si se cumplen las condiciones declaradas anteriormente en los tokens, se imprimirá un resultado exitoso a través de la variable mencionada anteriormente:

A continuación, se explica el significado de los indicadores Línea, Tipo, Valor, Posición:

- **Línea:** Se refiere al renglón en que fue ingresado el lexema.
- **Tipo:** El analizador considera el tipo de dato ingresado por el usuario (Identificador, Número, Suma, Resta, etc., etc.) que hacen referencia a los tokens.
- **Valor:** La función de este indicador es, sencillamente, repetir el lexema ingresado.

```
# Prueba de ingreso
def prueba(data):
    global resultado_lexema

    analizador = lex.lex()
    analizador.input(data)

    resultado_lexema.clear()
    while True:
        tok = analizador.token()
        if not tok:
            break
        # print("lexema de "+tok.type+" valor "+tok.value+" linea "+str(tok.lineno))
        estado = "Linea {:4} Tipo {:16} Valor {:16} Posicion {:4}".format(str(tok.lineno),str(tok.type) ,str(tok.value), str(tok.lexpos))
        resultado_lexema.append(estado)
    return resultado_lexema
```

- **Posición:** Es un contador donde se considera la posición del cursor en que se ingresa el valor.

```
# Prueba de ingreso
def prueba(data):
    global resultado_lexema

    analizador = lex.lex()
    analizador.input(data)

    resultado_lexema.clear()
    while True:
        tok = analizador.token()
        if not tok:
            break
        # print("lexema de "+tok.type+" valor "+tok.value+" linea "+str(tok.lineno))
        estado = "Linea {:4} Tipo {:16} Valor {:16} Posicion {:4}".format(str(tok.lineno),str(tok.type) ,str(tok.value), str(tok.lexpos) )
        resultado_lexema.append(estado)
    return resultado_lexema
```



Posteriormente, se instancia el analizador léxico para imprimir la indicación “ingrese” que permitirá al usuario ingresar los datos y mediante ello realizará todo el análisis valorando a través de cada uno de los tokens, como se indica en el paso anterior.

Por último, se manda llamar a la variable del resultado, dónde se alojará y mostrará el resultado obtenido del análisis léxico.

```
# instanciamos el analizador lexico
analizador = lex.lex()

if __name__ == '__main__':
    while True:
        data = input("ingrese: ")
        prueba(data)
        print(resultado_lexema)
```

Para concluir, realizamos una prueba, para demostrar el funcionamiento de este analizador léxico:

```
Ingrese: hola
['Linea 1  Tipo IDENTIFICADOR  Valor hola          Posicion 0  ']
Ingrese: cómo estás
['Linea 1  Tipo IDENTIFICADOR  Valor cómo           Posicion 0  ', 'Linea 1  Tipo IDENTIFICADOR  Valor estás         Posicion 5  ']
Ingrese: Ok, amigo
['Linea 1  Tipo IDENTIFICADOR  Valor Ok            Posicion 0  ', 'Linea 1  Tipo COMA             Valor ,            Posicion 2  ', 'Linea 1  Tipo IDENTIFI
Ingrese: 15.2
['Linea 1  Tipo ENTERO          Valor 15           Posicion 0  ', 'Linea 1  Tipo PUNTO            Valor .            Posicion 2  ', 'Linea 1  Tipo ENTERO          Valor 15           Posicion 2  ', 'Linea 1  Tipo IDENTIFI
Ingrese: 2 + 5 = 7
['Linea 1  Tipo ENTERO          Valor 2            Posicion 0  ', 'Linea 1  Tipo SUMA             Valor +            Posicion 2  ', 'Linea 1  Tipo ENTERO          Valor 7             Posicion 2  ', 'Linea 1  Tipo ENTERO          Valor 2            Posicion 8  ']
```