

2017



LINUX BASH EN WINDOWS 10

FUNDAMENTOS DE PROGRAMACIÓN

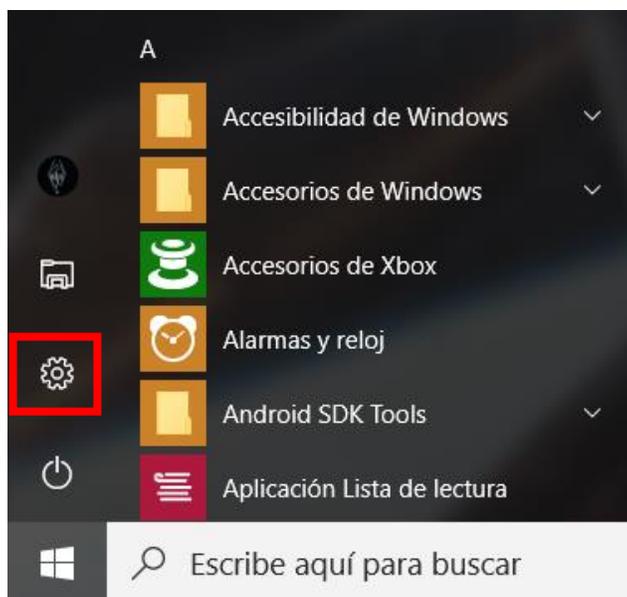
Con la salida de la actualización Anniversary para Windows 10 se permite a los desarrolladores la opción para instalar el Bash de Linux dentro de Windows.

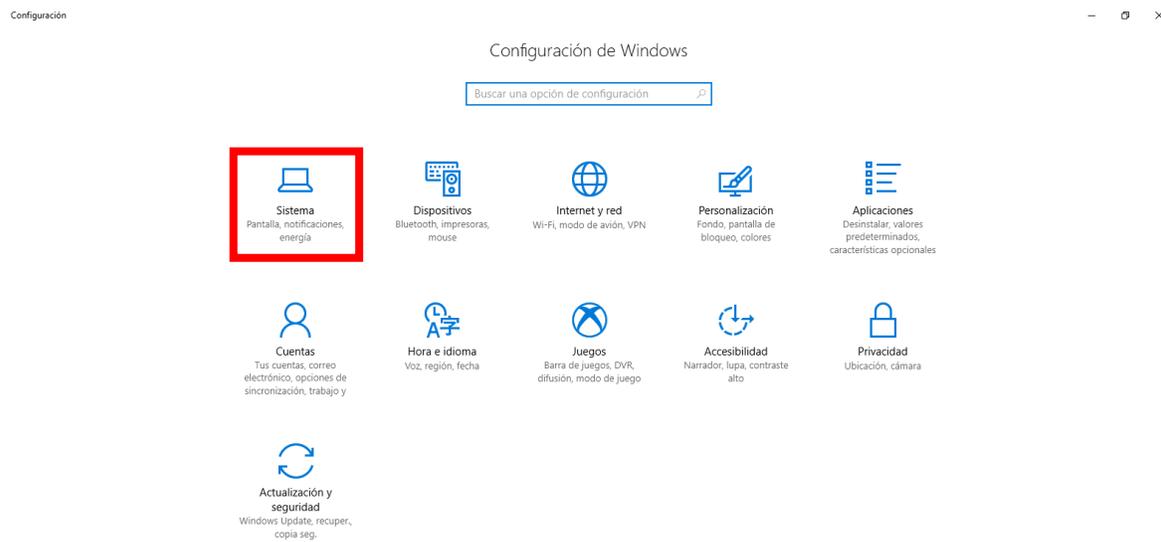
1. Requisitos

- ✓ Conexión a internet
- ✓ Windows 10 Anniversary Update
- ✓ Al menos 2 GB disponibles de disco duro

2. Verificar nuestra versión de Windows

Para poder instalar el *Bash de Linux* necesitamos tener instalada la actualización *Anniversary*, para poder comprobarlo abrimos el *Panel de Configuración*.



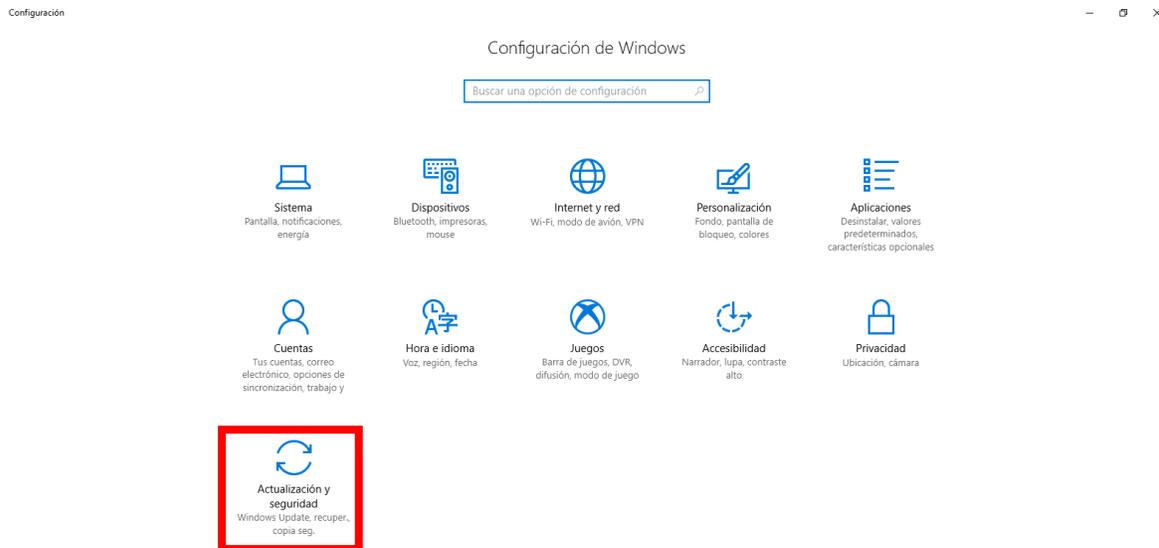
Seleccionamos *Sistema*.

Seleccionamos *Acerca de* y verificamos el apartado **Versión**, esta deberá de ser *igual o mayor a 1607* (Build de Anniversary Update), como podemos observar la versión actual del equipo es 1703 (Build de Creators Update) y puedo seguir con el siguiente paso, *en caso contrario debemos de actualizar nuestro Windows*.

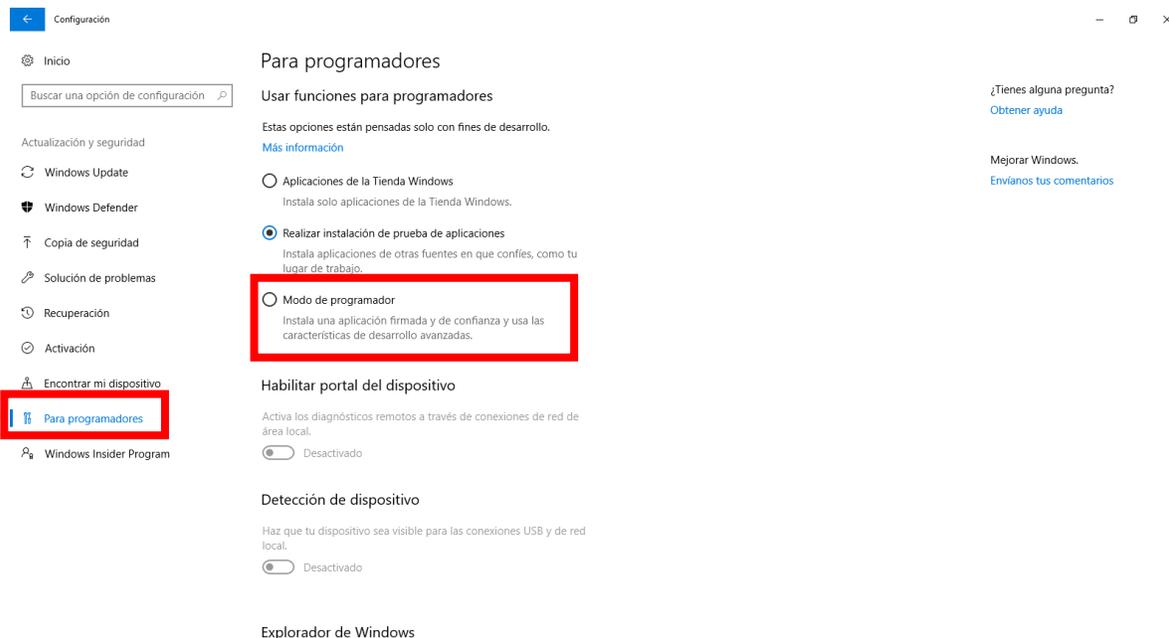


3. Habilitar el Modo Programador

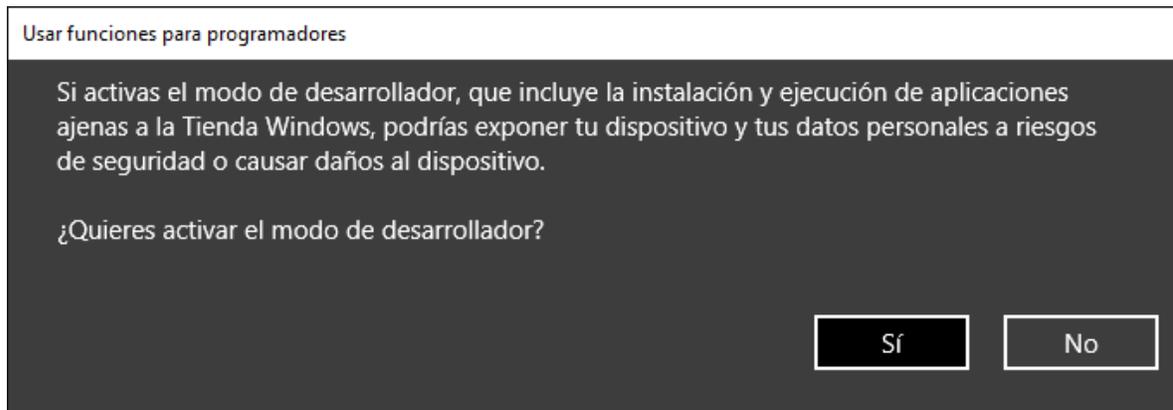
Para habilitar el modo programador en el *Panel de Configuración* seleccionamos *Actualización y seguridad*.



Seleccionamos la opción *Para programadores* y seleccionamos la opción **Modo de programador**.



Aparecerá la siguiente ventana y damos clic en *Sí*.



Iniciará el proceso de descarga e instalación del paquete de programador.



Configuración

Inicio

Buscar una opción de configuración

Actualización y seguridad

- Windows Update
- Windows Defender
- Copia de seguridad
- Solución de problemas
- Recuperación
- Activación
- Encontrar mi dispositivo
- Para programadores**
- Windows Insider Program

Para programadores

Usar funciones para programadores

Estas opciones están pensadas solo con fines de desarrollo.

[Más información](#)

- Aplicaciones de la Tienda Windows
Instala solo aplicaciones de la Tienda Windows.
- Realizar instalación de prueba de aplicaciones
Instala aplicaciones de otras fuentes en que confíes, como tu lugar de trabajo.
- Modo de programador**
Instala una aplicación firmada y de confianza y usa las características de desarrollo avanzadas.
Instalando el paquete de modo de programador

Habilitar portal del dispositivo

Activa los diagnósticos remotos a través de conexiones de red de área local.

Desactivado

Detección de dispositivo

Haz que tu dispositivo sea visible para las conexiones USB y de red local.

Desactivado

¿Tienes alguna pregunta?
[Obtener ayuda](#)

Mejorar Windows.
[Envíanos tus comentarios](#)

Configuración

Inicio

Buscar una opción de configuración

Actualización y seguridad

- Windows Update
- Windows Defender
- Copia de seguridad
- Solución de problemas
- Recuperación
- Activación
- Encontrar mi dispositivo
- Para programadores**
- Windows Insider Program

Para programadores

Usar funciones para programadores

Estas opciones están pensadas solo con fines de desarrollo.

[Más información](#)

- Aplicaciones de la Tienda Windows
Instala solo aplicaciones de la Tienda Windows.
- Realizar instalación de prueba de aplicaciones
Instala aplicaciones de otras fuentes en que confíes, como tu lugar de trabajo.
- Modo de programador**
Instala una aplicación firmada y de confianza y usa las características de desarrollo avanzadas.
Se instaló el paquete de modo de desarrollador. Las herramientas remotas para escritorio ahora están habilitadas.

Habilitar portal del dispositivo

Activa los diagnósticos remotos a través de conexiones de red de área local.

Desactivado

Detección de dispositivo

Haz que tu dispositivo sea visible para las conexiones USB y de red local.

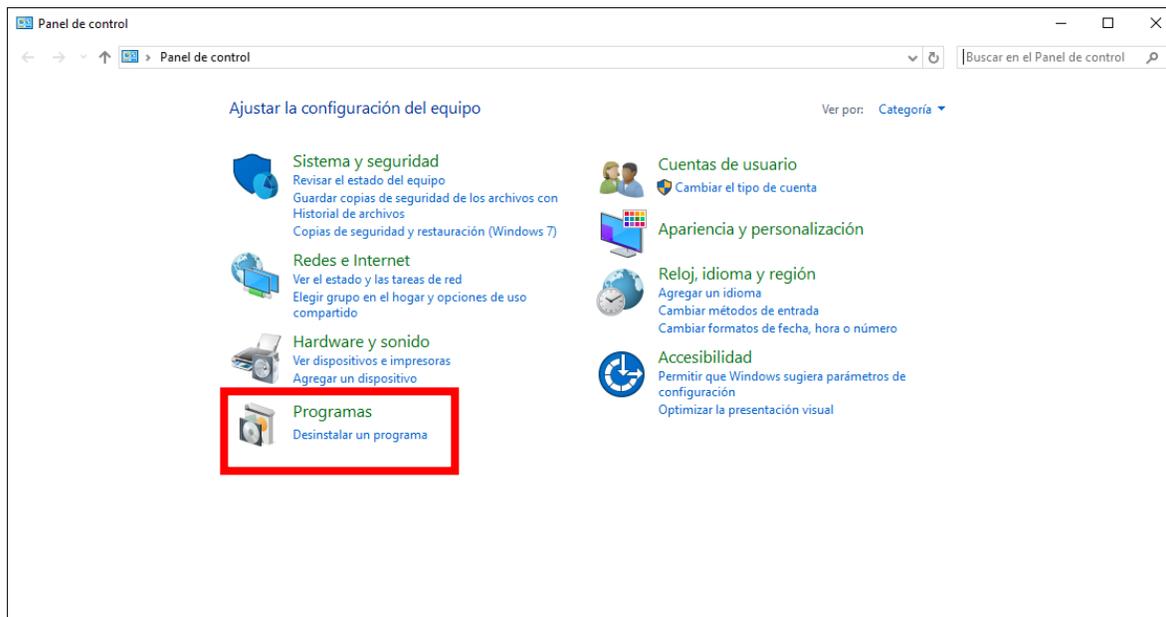
Desactivado

¿Tienes alguna pregunta?
[Obtener ayuda](#)

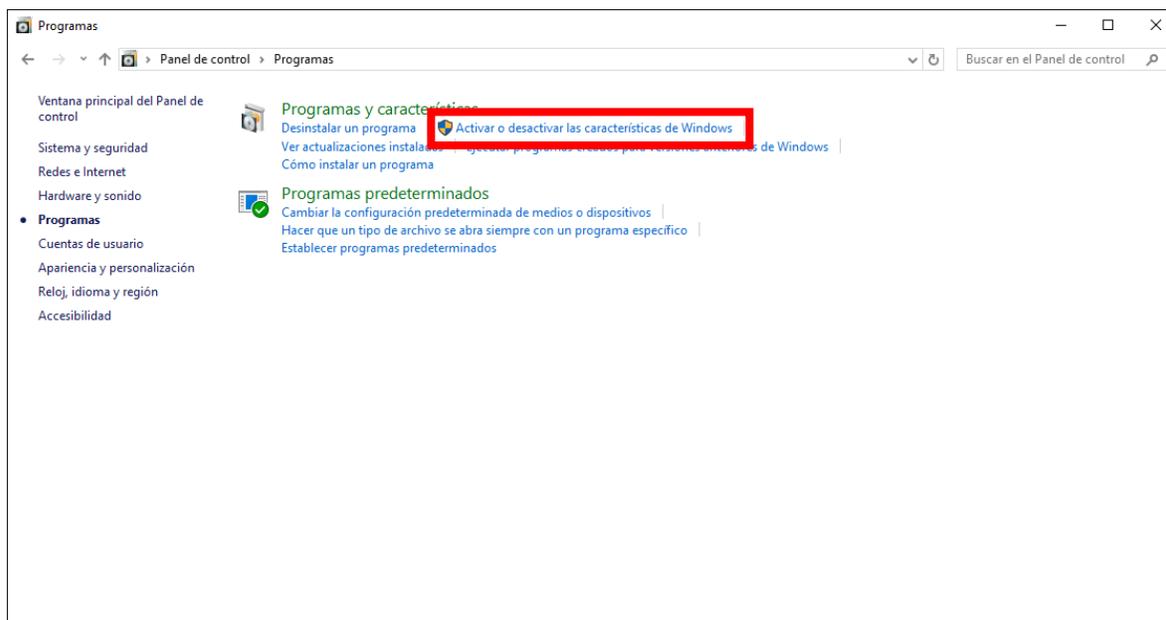
Mejorar Windows.
[Envíanos tus comentarios](#)

4. Activar el Subsistema de Windows para Linux

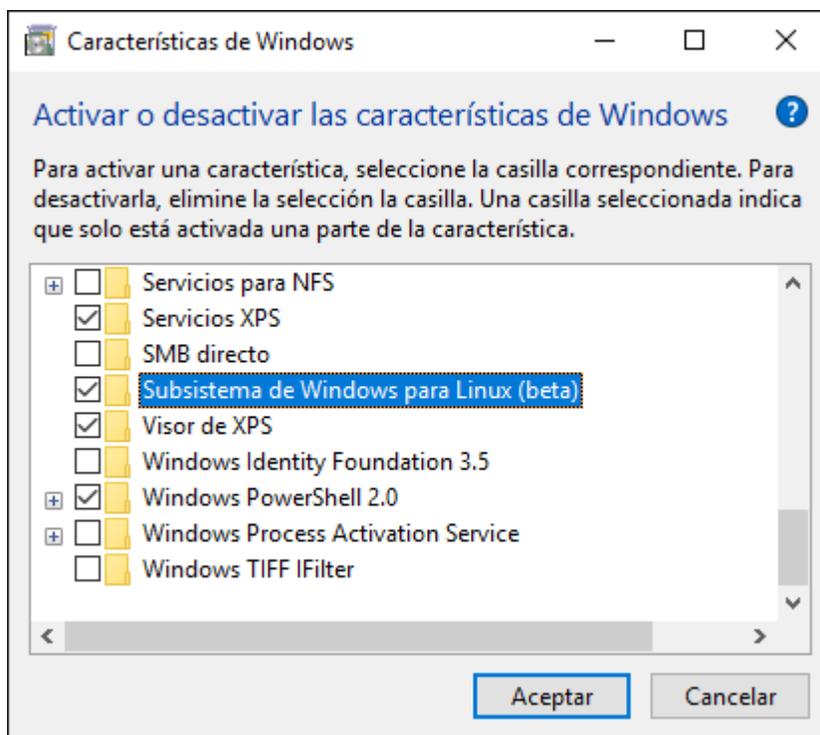
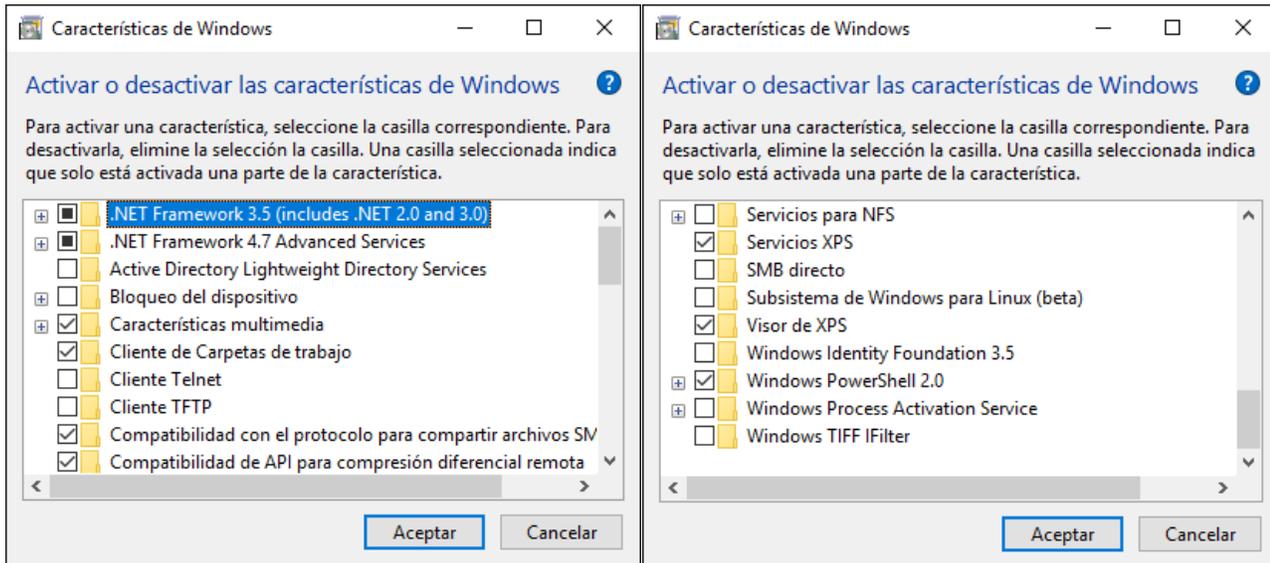
Abrimos el *Panel de Control* y damos clic en **Programas**.



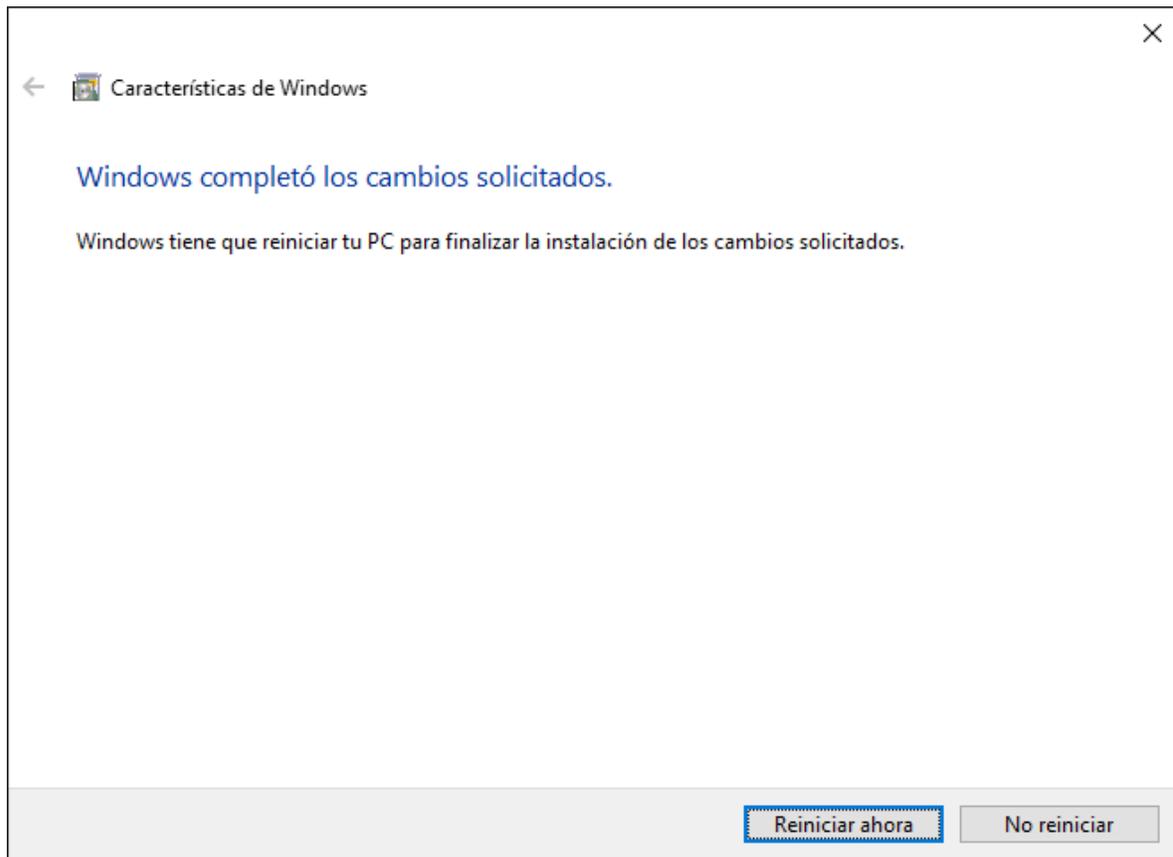
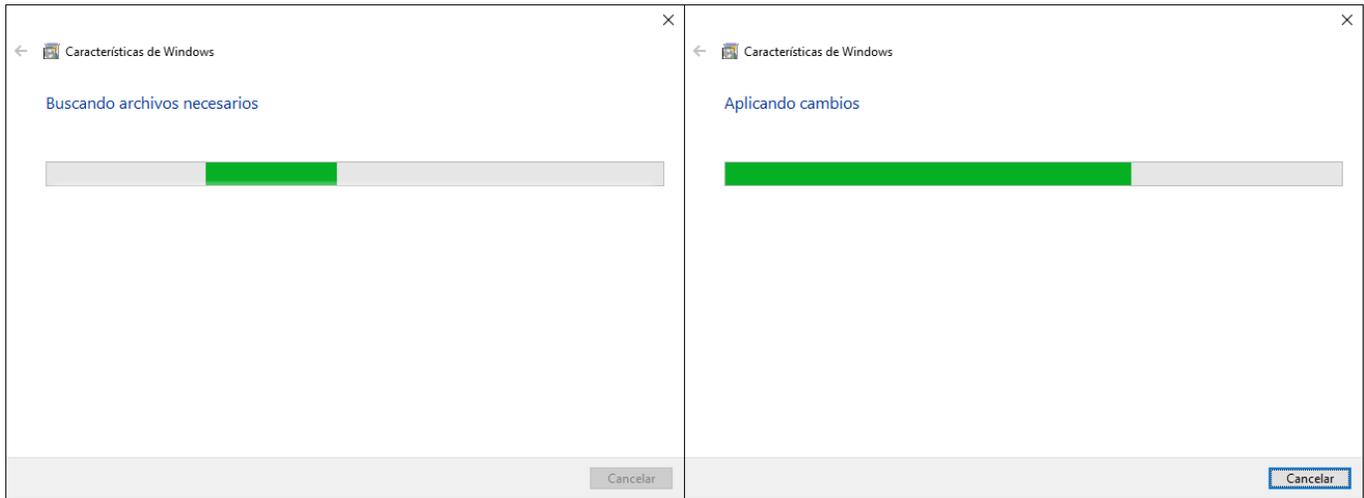
Damos clic en *Activar o desactivar características de Windows*.



Aparecerá la siguiente pantalla, bajamos hasta encontrar **Subsistema de Windows para Linux (beta)**, lo seleccionamos y damos clic en *Aceptar*.

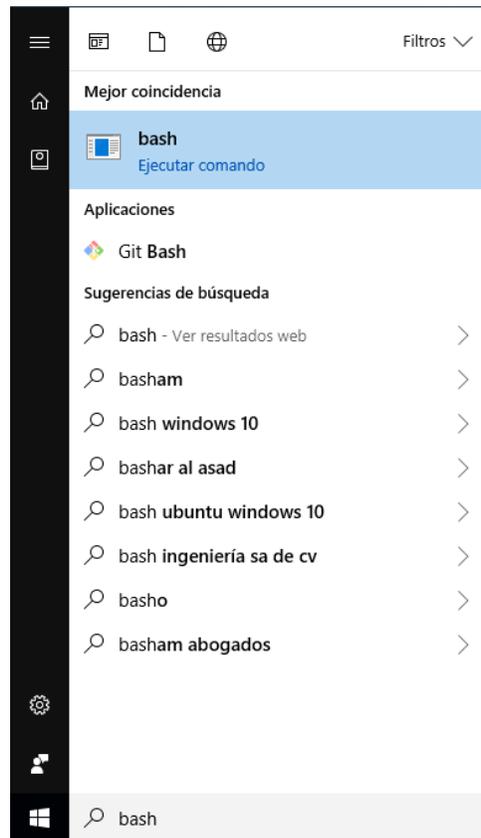


Esperamos a que termine de descargar y configurar los archivos, nos pedirá reiniciar, damos clic en **Reiniciar Ahora**.



5. Instalar el Bash de Linux

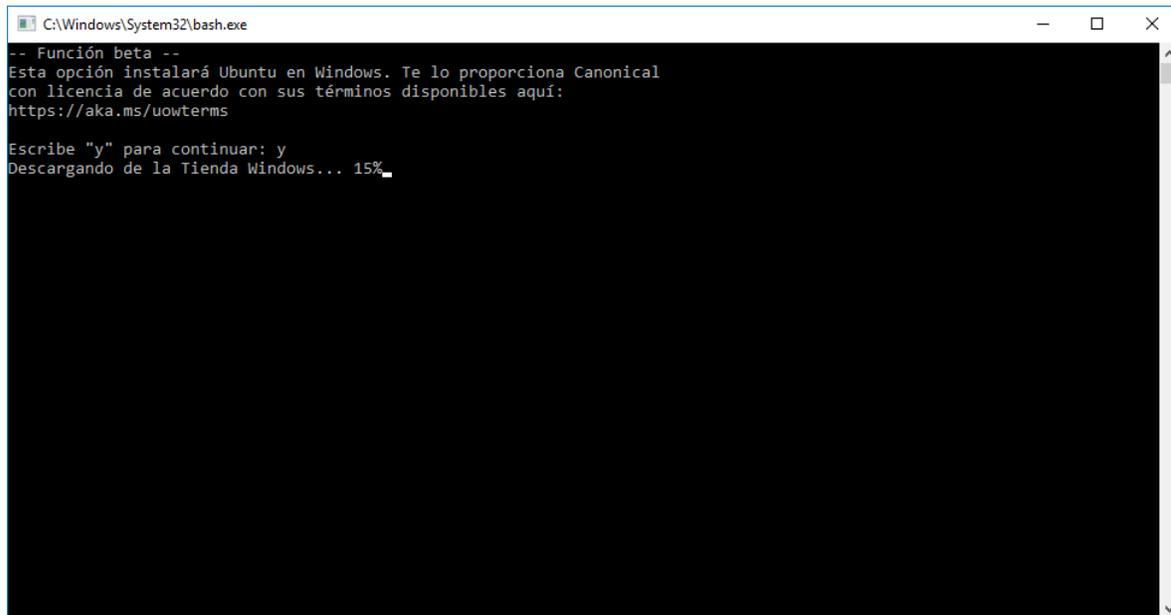
En el Buscador escribimos *Bash* y seleccionamos el archivo que aparece.



Aparecerá la siguiente ventana, nos pedirá que aceptemos los *términos y condiciones del programa*, escribimos *y*, presionamos *Enter*.

```
C:\Windows\System32\bash.exe
-- Función beta --
Esta opción instalará Ubuntu en Windows. Te lo proporciona Canonical
con licencia de acuerdo con sus términos disponibles aquí:
https://aka.ms/uowterms
Escribe "y" para continuar:
```

Iniciará la descarga.

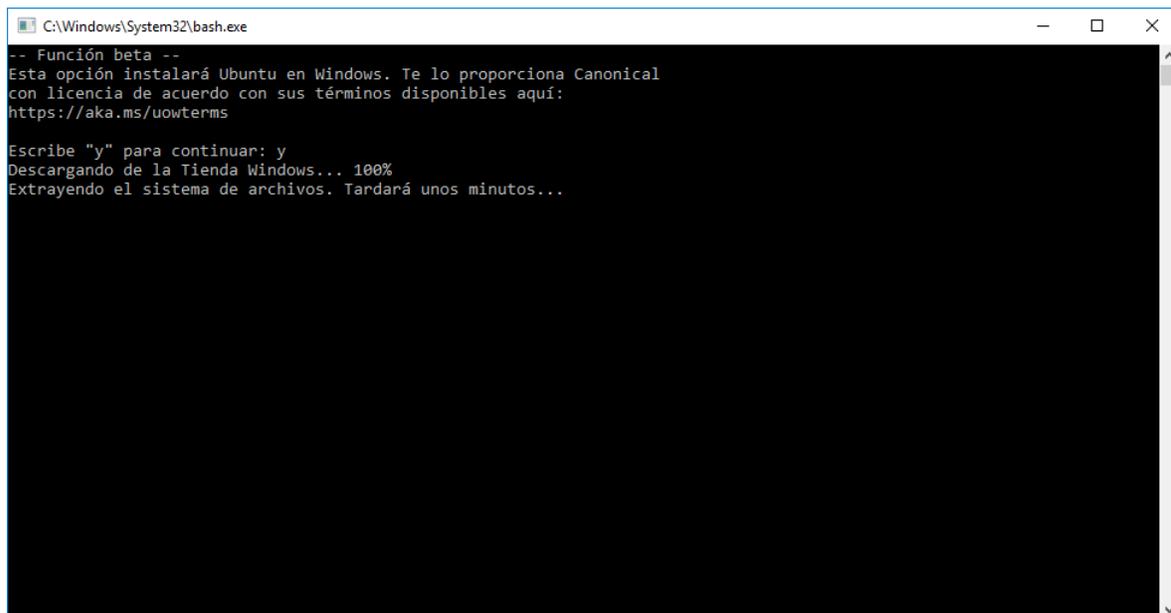


```
C:\Windows\System32\bash.exe
-- Función beta --
Esta opción instalará Ubuntu en Windows. Te lo proporciona Canonical
con licencia de acuerdo con sus términos disponibles aquí:
https://aka.ms/uowterms

Escribe "y" para continuar: y
Descargando de la Tienda Windows... 15%
```

A continuación, puede suceder uno de los siguientes dos casos:

- Parece que no responde el programa, pero como no tengo una cuenta asociada a mi computadora no me pedirá iniciar sesión con una cuenta de Windows, en este caso debemos de presionar *Enter* para que aparezca el resto de configuraciones y escribir *y* para continuar.



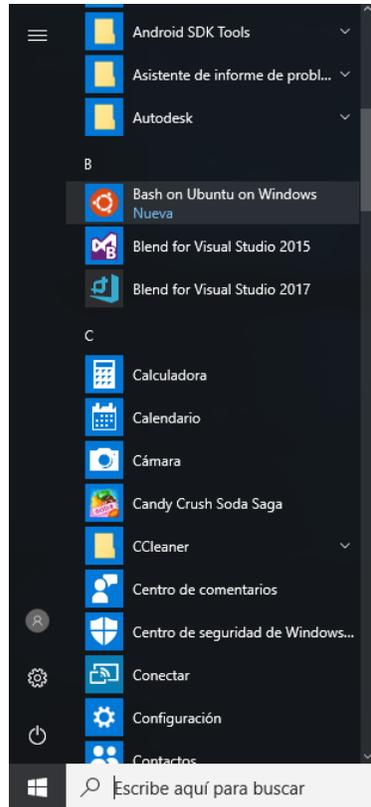
```
C:\Windows\System32\bash.exe
-- Función beta --
Esta opción instalará Ubuntu en Windows. Te lo proporciona Canonical
con licencia de acuerdo con sus términos disponibles aquí:
https://aka.ms/uowterms

Escribe "y" para continuar: y
Descargando de la Tienda Windows... 100%
Extrayendo el sistema de archivos. Tardará unos minutos...
```

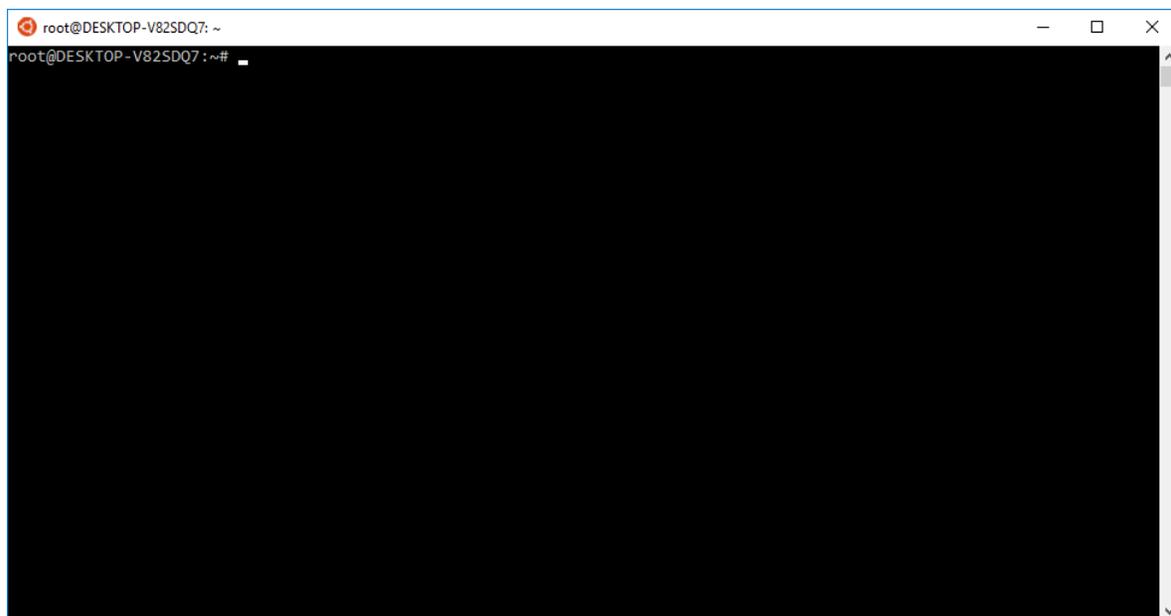
- Aparecerá un cuadro de dialogo pidiéndonos iniciar sesión para bajar la aplicación desde la tienda de Windows, esto sucede cuando tenemos asociada nuestra cuenta en nuestra computadora, iniciamos sesión y dejamos que el programa continúe.

6. Configurar GCC en el Bash de Ubuntu

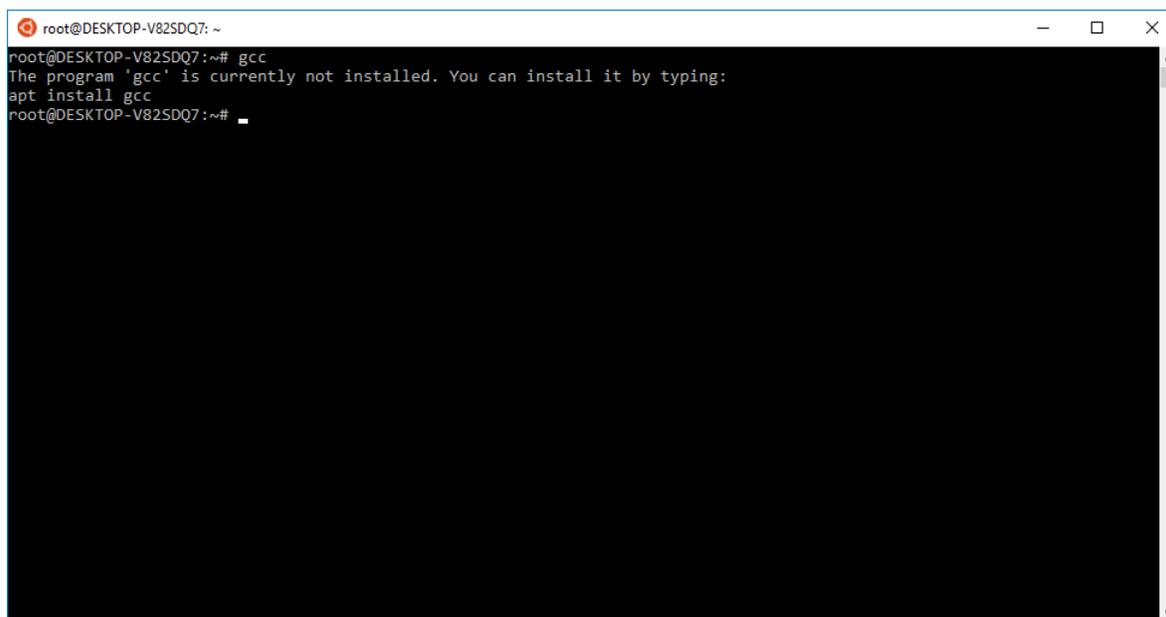
En inicio buscamos en la **letra B** *Bash on Ubuntu on Windows* y le damos clic.



Nos abrirá la terminal



Verificamos si tenemos instalado GCC escribiendo gcc y presionando *Enter*.

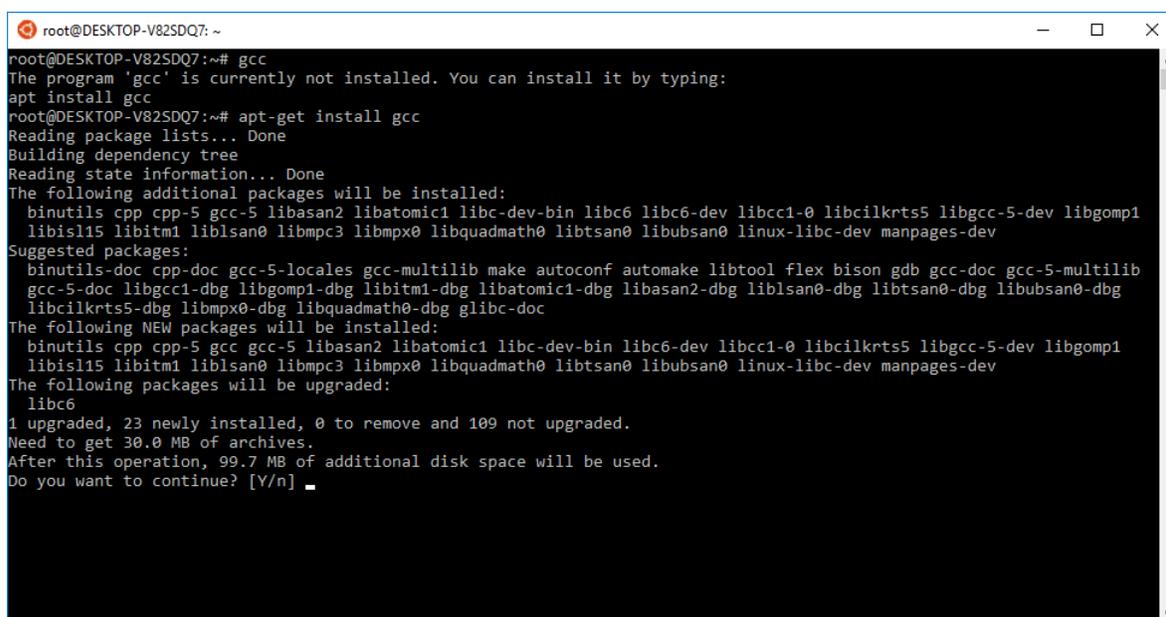


```
root@DESKTOP-V82SDQ7: ~  
root@DESKTOP-V82SDQ7:~# gcc  
The program 'gcc' is currently not installed. You can install it by typing:  
apt install gcc  
root@DESKTOP-V82SDQ7:~#
```

Como podemos ver no lo tenemos instalado, ejecutamos el siguiente comando:

apt-get install gcc

Presionamos *Enter*, nos pedirá confirmar la instalación, presionamos **Y** y presionamos *Enter*.



```
root@DESKTOP-V82SDQ7: ~  
root@DESKTOP-V82SDQ7:~# gcc  
The program 'gcc' is currently not installed. You can install it by typing:  
apt install gcc  
root@DESKTOP-V82SDQ7:~# apt-get install gcc  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  binutils cpp cpp-5 gcc-5 libasan2 libatomic1 libc-dev-bin libc6 libc6-dev libcc1-0 libcilkrts5 libgcc-5-dev libgomp1  
  libisl15 libitm1 liblsan0 libmpc3 libmpx0 libquadmath0 libtsan0 libubsan0 linux-libc-dev manpages-dev  
Suggested packages:  
  binutils-doc cpp-doc gcc-5-locales gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-5-multilib  
  gcc-5-doc libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan2-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg  
  libcilkrts5-dbg libmpx0-dbg libquadmath0-dbg glibc-doc  
The following NEW packages will be installed:  
  binutils cpp cpp-5 gcc gcc-5 libasan2 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libgcc-5-dev libgomp1  
  libisl15 libitm1 liblsan0 libmpc3 libmpx0 libquadmath0 libtsan0 libubsan0 linux-libc-dev manpages-dev  
The following packages will be upgraded:  
  libc6  
  1 upgraded, 23 newly installed, 0 to remove and 109 not upgraded.  
Need to get 30.0 MB of archives.  
After this operation, 99.7 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Iniciará la descarga y esperamos a que termine la instalación.

```
root@DESKTOP-V82SDQ7: ~  
root@DESKTOP-V82SDQ7:~# gcc  
The program 'gcc' is currently not installed. You can install it by typing:  
apt install gcc  
root@DESKTOP-V82SDQ7:~# apt-get install gcc  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  binutils cpp cpp-5 gcc-5 libasan2 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libgcc-5-dev libgomp1  
  libisl15 libitm1 liblsan0 libmpc3 libmpx0 libquadmath0 libtsan0 libubsan0 linux-libc-dev manpages-dev  
Suggested packages:  
  binutils-doc cpp-doc gcc-5-locales gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-5-multilib  
  gcc-5-doc libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan2-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg  
  libcilkrts5-dbg libmpx0-dbg libquadmath0-dbg glibc-doc  
The following NEW packages will be installed:  
  binutils cpp cpp-5 gcc gcc-5 libasan2 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libgcc-5-dev libgomp1  
  libisl15 libitm1 liblsan0 libmpc3 libmpx0 libquadmath0 libtsan0 libubsan0 linux-libc-dev manpages-dev  
The following packages will be upgraded:  
  libc6  
1 upgraded, 23 newly installed, 0 to remove and 109 not upgraded.  
Need to get 30.0 MB of archives.  
After this operation, 99,7 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libc6 amd64 2.23-0ubuntu9 [2,586 kB]  
6% [1 libc6 2,079 kB/2,586 kB 80%]
```

```
gcc-5-doc libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan2-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg  
  libcilkrts5-dbg libmpx0-dbg libquadmath0-dbg glibc-doc  
The following NEW packages will be installed:  
  binutils cpp cpp-5 gcc gcc-5 libasan2 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libgcc-5-dev libgomp1  
  libisl15 libitm1 liblsan0 libmpc3 libmpx0 libquadmath0 libtsan0 libubsan0 linux-libc-dev manpages-dev  
The following packages will be upgraded:  
  libc6  
1 upgraded, 23 newly installed, 0 to remove and 109 not upgraded.  
Need to get 30.0 MB of archives.  
After this operation, 99,7 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libc6 amd64 2.23-0ubuntu9 [2,586 kB]  
Get:2 http://archive.ubuntu.com/ubuntu xenial/main amd64 libmpc3 amd64 1.0.3-1 [39,7 kB]  
Get:3 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 binutils amd64 2.26.1-1ubuntu1~16.04.4 [2,311 kB]  
Get:4 http://archive.ubuntu.com/ubuntu xenial/main amd64 libisl15 amd64 0.16.1-1 [524 kB]  
Get:5 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 cpp-5 amd64 5.4.0-6ubuntu1~16.04.4 [7,653 kB]  
Get:6 http://archive.ubuntu.com/ubuntu xenial/main amd64 cpp amd64 4:5.3.1-1ubuntu1 [27,7 kB]  
Get:7 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libcc1-0 amd64 5.4.0-6ubuntu1~16.04.4 [38,8 kB]  
Get:8 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libgomp1 amd64 5.4.0-6ubuntu1~16.04.4 [55,0 kB]  
Get:9 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libitm1 amd64 5.4.0-6ubuntu1~16.04.4 [27,4 kB]  
Get:10 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libatomic1 amd64 5.4.0-6ubuntu1~16.04.4 [8,912 B]  
Get:11 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libasan2 amd64 5.4.0-6ubuntu1~16.04.4 [264 kB]  
Get:12 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 liblsan0 amd64 5.4.0-6ubuntu1~16.04.4 [105 kB]  
Get:13 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libtsan0 amd64 5.4.0-6ubuntu1~16.04.4 [244 kB]  
Get:14 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libubsan0 amd64 5.4.0-6ubuntu1~16.04.4 [95,3 kB]  
Get:15 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libcilkrts5 amd64 5.4.0-6ubuntu1~16.04.4 [40,1 kB]  
Get:16 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libmpx0 amd64 5.4.0-6ubuntu1~16.04.4 [9,766 B]  
Get:17 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libquadmath0 amd64 5.4.0-6ubuntu1~16.04.4 [131 kB]  
Get:18 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libgcc-5-dev amd64 5.4.0-6ubuntu1~16.04.4 [2,237 kB]  
53% [18 libgcc-5-dev 446 kB/2,237 kB 20%] 429 kB/s 35s
```

```
root@DESKTOP-V82SDQ7: ~
Selecting previously unselected package manpages-dev.
Preparing to unpack .../manpages-dev_4.04-2_all.deb ...
Unpacking manpages-dev (4.04-2) ...
Processing triggers for libc-bin (2.23-0ubuntu7) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libmpc3:amd64 (1.0.3-1) ...
Setting up binutils (2.26.1-1ubuntu1~16.04.4) ...
Setting up libisl15:amd64 (0.16.1-1) ...
Setting up cpp-5 (5.4.0-6ubuntu1~16.04.4) ...
Setting up cpp (4:5.3.1-1ubuntu1) ...
Setting up libcc1-0:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libgomp1:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libitm1:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libatomic1:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libasan2:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up liblsan0:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libtsan0:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libubsan0:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libcilkrts5:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libmpx0:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libquadmath0:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up libgcc-5-dev:amd64 (5.4.0-6ubuntu1~16.04.4) ...
Setting up gcc-5 (5.4.0-6ubuntu1~16.04.4) ...
Setting up gcc (4:5.3.1-1ubuntu1) ...
Setting up libc-dev-bin (2.23-0ubuntu9) ...
Setting up linux-libc-dev:amd64 (4.4.0-92.115) ...
Setting up libc6-dev:amd64 (2.23-0ubuntu9) ...
Setting up manpages-dev (4.04-2) ...
Processing triggers for libc-bin (2.23-0ubuntu7) ...
root@DESKTOP-V82SDQ7:~#
```

7. Verificar si GCC está correctamente instalado

Ahora necesitamos crear un programa para verificar si GCC funciona adecuadamente, *creamos un nuevo archivo de texto* y copiamos el siguiente código:

```
#include <stdio.h>

int main(){
    printf("GCC funciona correctamente!\n");
    return 0;
}
```

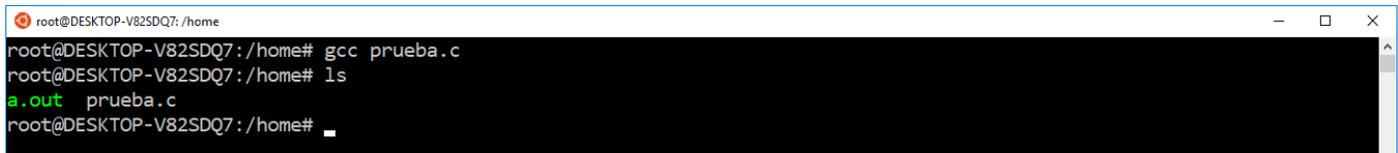
Lo guardamos con extensión .c, por ejemplo *prueba.c*

```
root@DESKTOP-V82SDQ7: /home
root@DESKTOP-V82SDQ7:/home# ls
prueba.c
root@DESKTOP-V82SDQ7:/home#
```

Escribimos **gcc prueba.c**

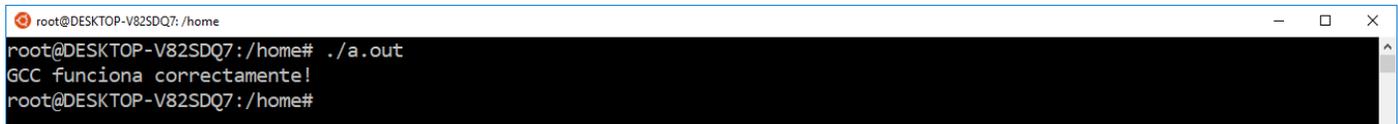
```
root@DESKTOP-V82SDQ7: /home
root@DESKTOP-V82SDQ7:/home# gcc prueba.c
root@DESKTOP-V82SDQ7:/home#
```

Parece que no pasó nada, pero si revisamos nuestros archivos veremos que tenemos un archivo llamado *a.out*, lo que acaba de suceder es que GCC compiló nuestro programa y no encontró ningún error.



```
root@DESKTOP-V82SDQ7: /home
root@DESKTOP-V82SDQ7: /home# gcc prueba.c
root@DESKTOP-V82SDQ7: /home# ls
a.out prueba.c
root@DESKTOP-V82SDQ7: /home#
```

Ahora ejecutamos el programa escribiendo **./a.out**



```
root@DESKTOP-V82SDQ7: /home
root@DESKTOP-V82SDQ7: /home# ./a.out
GCC funciona correctamente!
root@DESKTOP-V82SDQ7: /home#
```

Si aparece el mensaje “GCC funciona correctamente!” ya podremos empezar a programar en Lenguaje C.