|  |  |
| --- | --- |
| Facultad de Ingeniería | Laboratorio de docencia |

Laboratorios de computación

salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 20

No de práctica: 06

Integrantes: Leonardo Ponce Vivas

No de lista o brigada: 40

Semestre: 2021-1

Fecha de entrega: 17 de noviembre

Observaciones:

CALIFICACION:

Entorno de C (editores, compilación y ejecución)

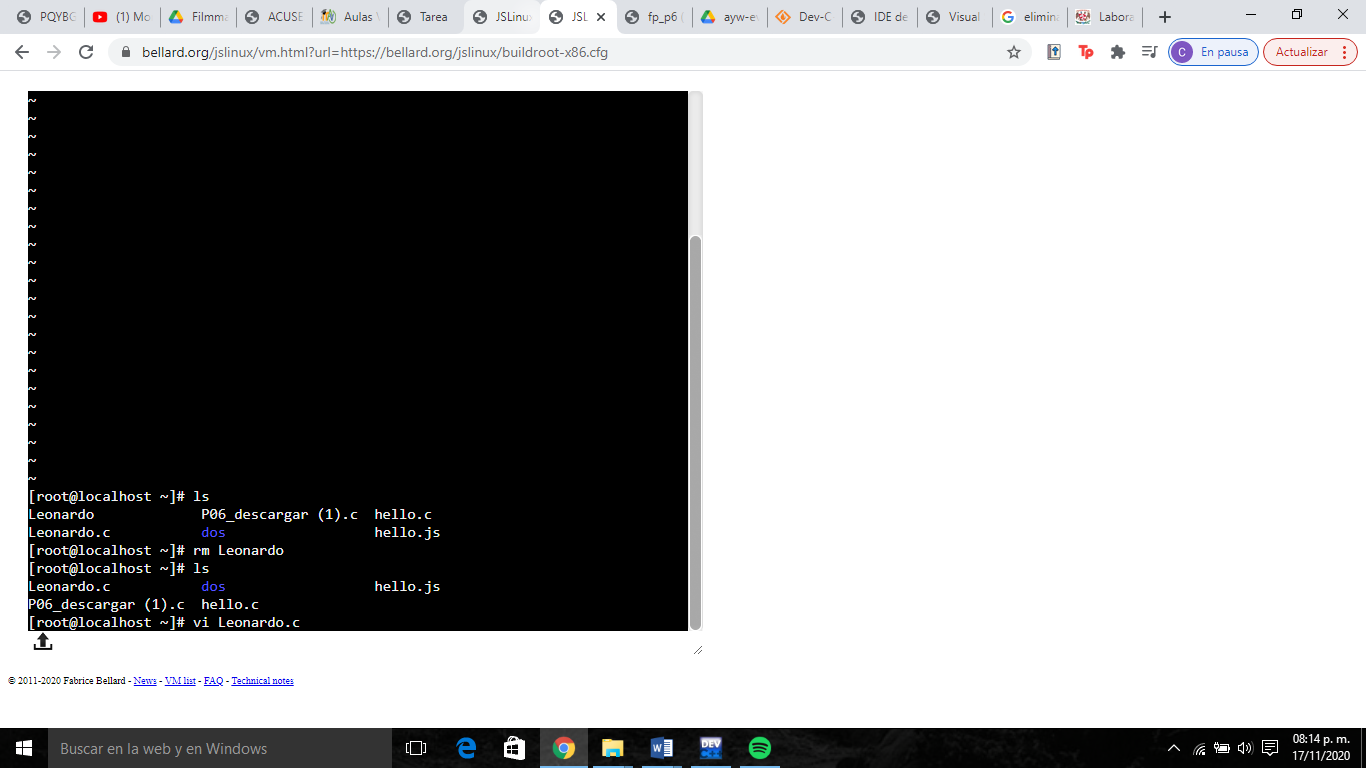
**Objetico:**

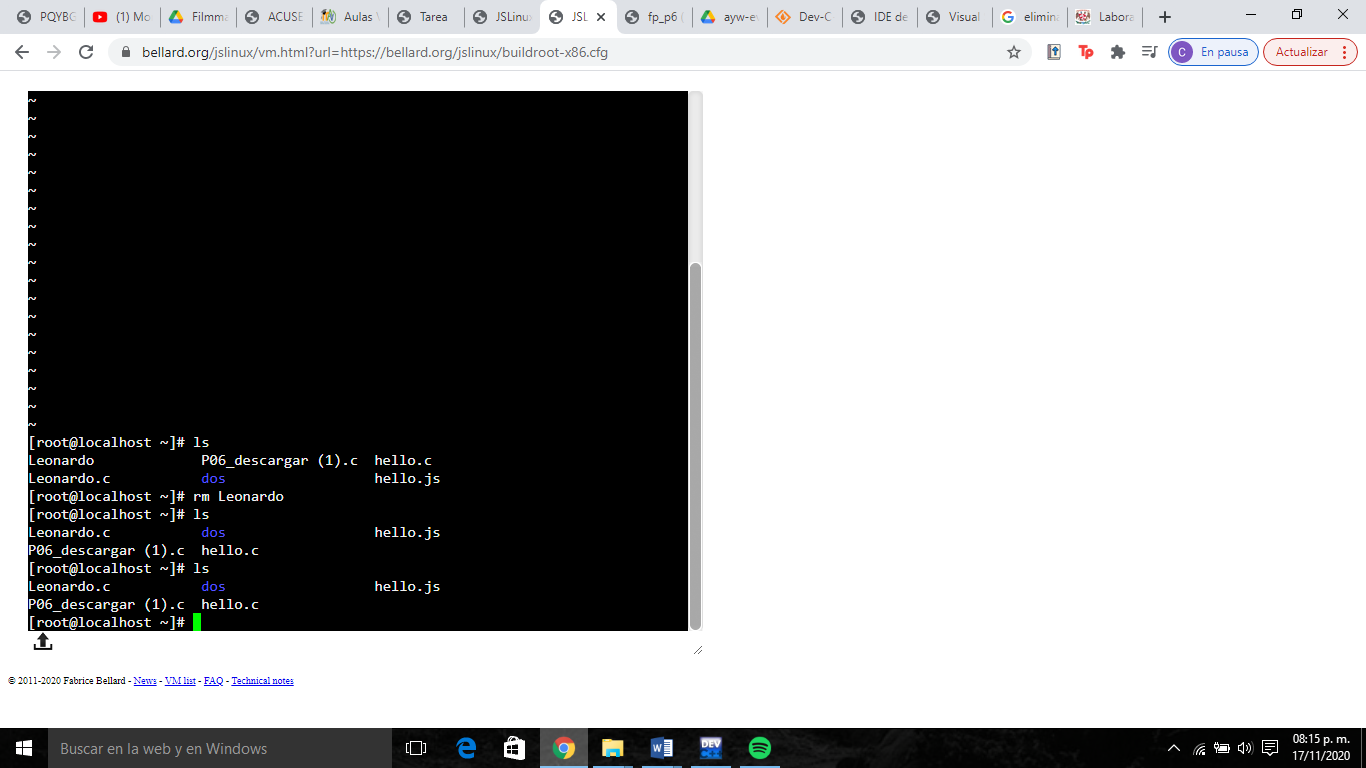
Elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

**Desarrollo:**

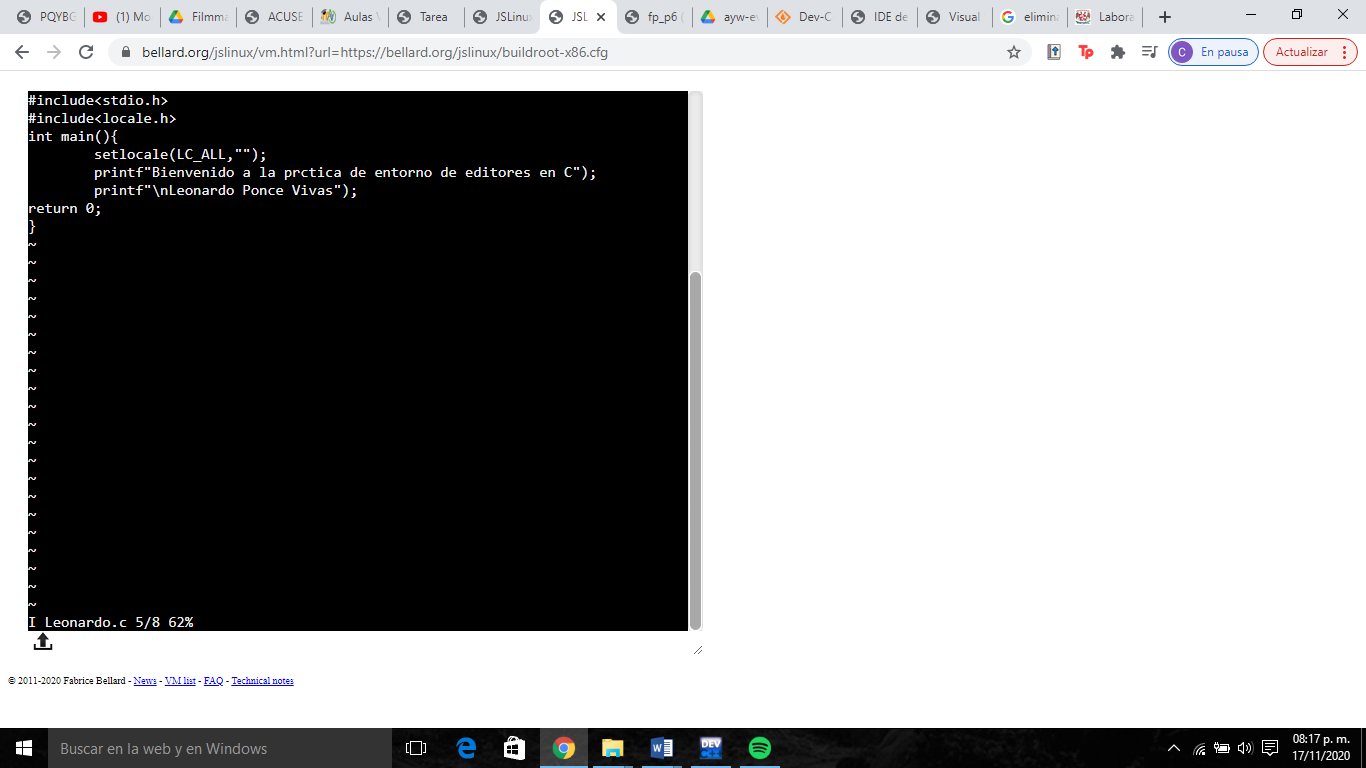
Un lenguaje de programación permite expresar una serie de instrucciones que podrán ser realizadas por una computadora, muy utilizado ya que la forma de dar instrucciones es muy cercana a lo que un humano podría abstraer, para ello se necesitaría conocer a fondo el microprocesador, el sistema operativo entre otros aspectos, el cual es C, conocido como un lenguaje de alto nivel, del cual las instrucciones podrían ser entendidas fácilmente por un humano.

*Crear un archivo de texto*

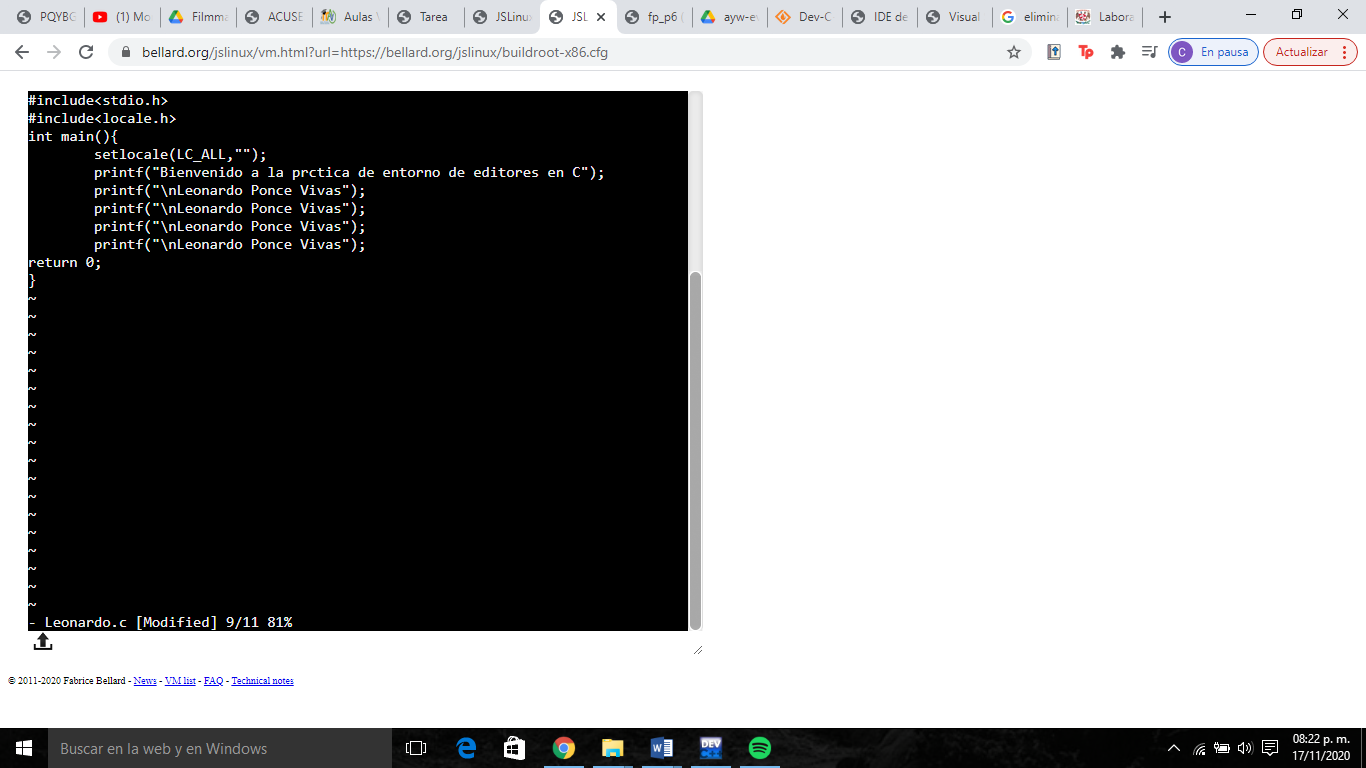




*Modificar un archivo existente*

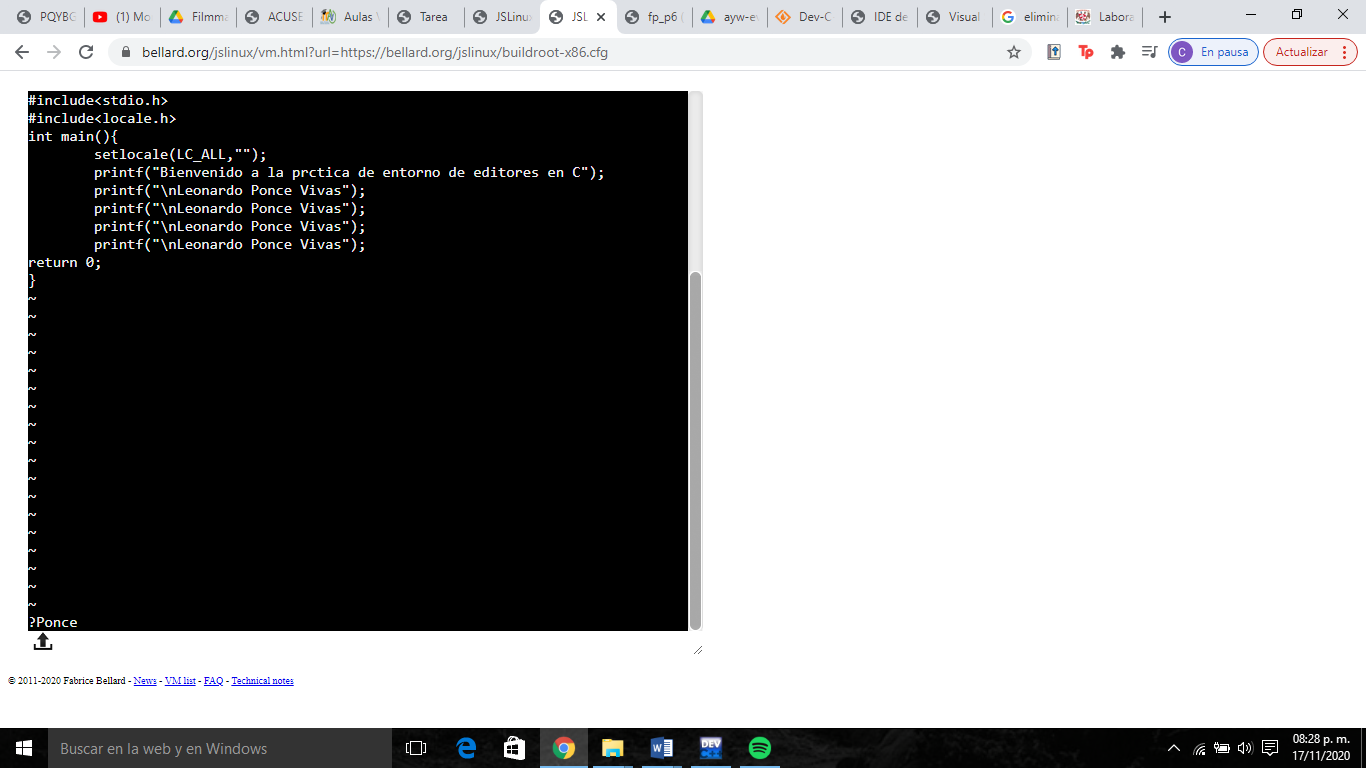


*Actualizar un archivo existente*

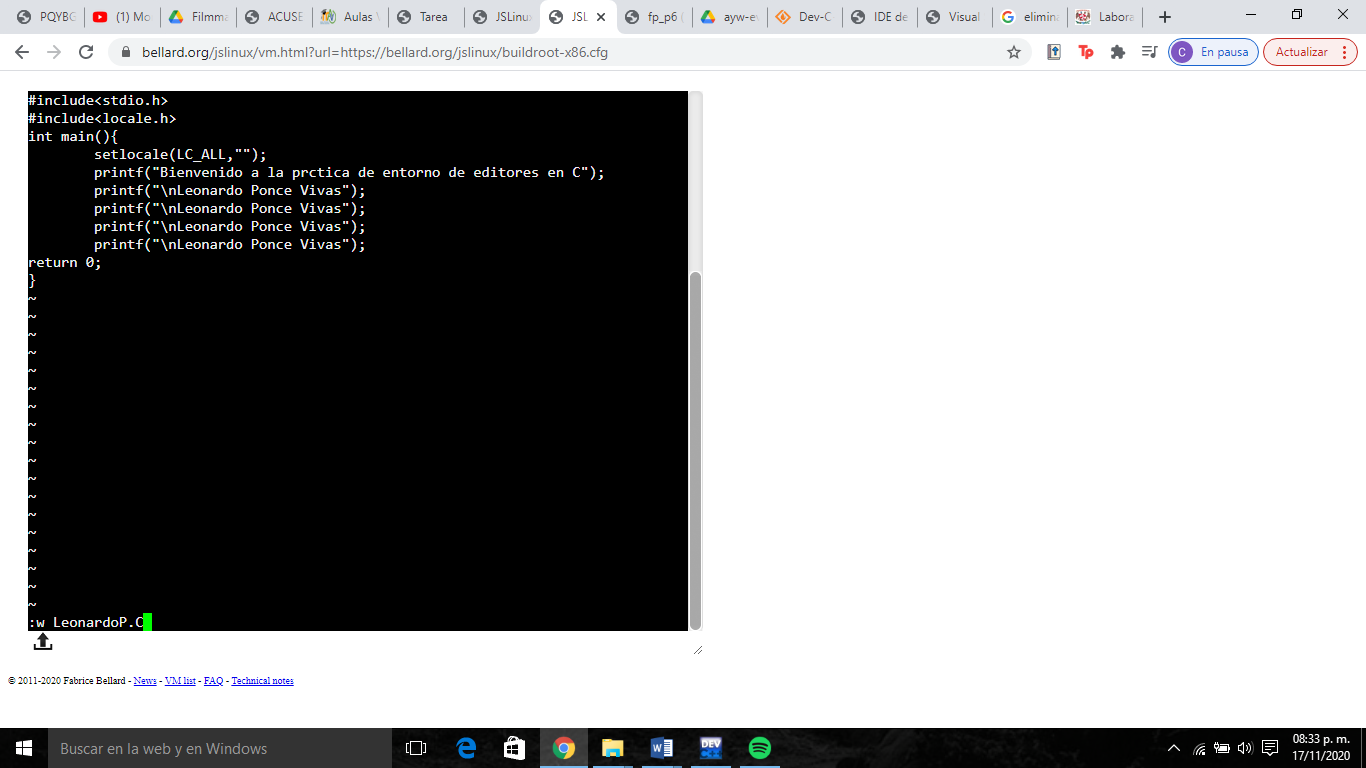


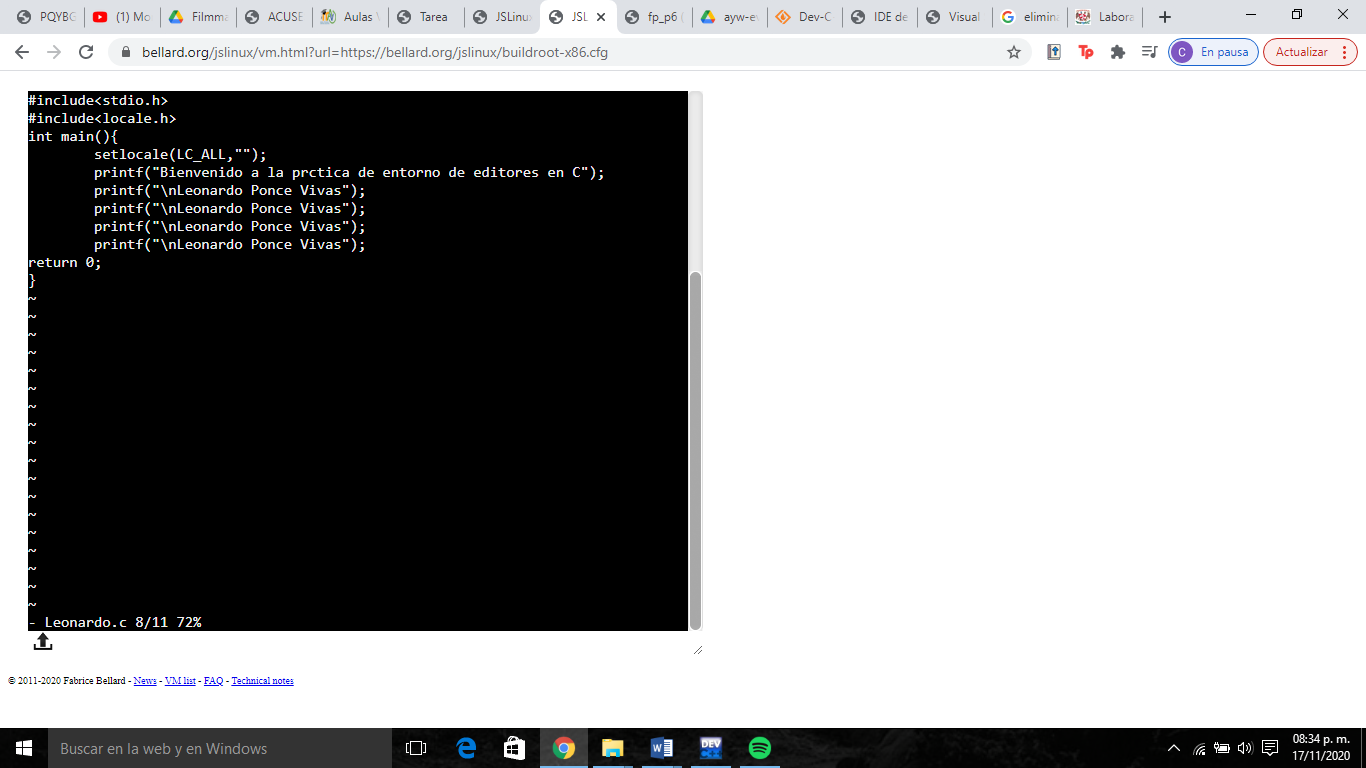
Se copió printf(“\nLeonardo Ponce Vivas”); - usando “yy” para copiar y “p” para pegar

Se corrigió errores en la escritura

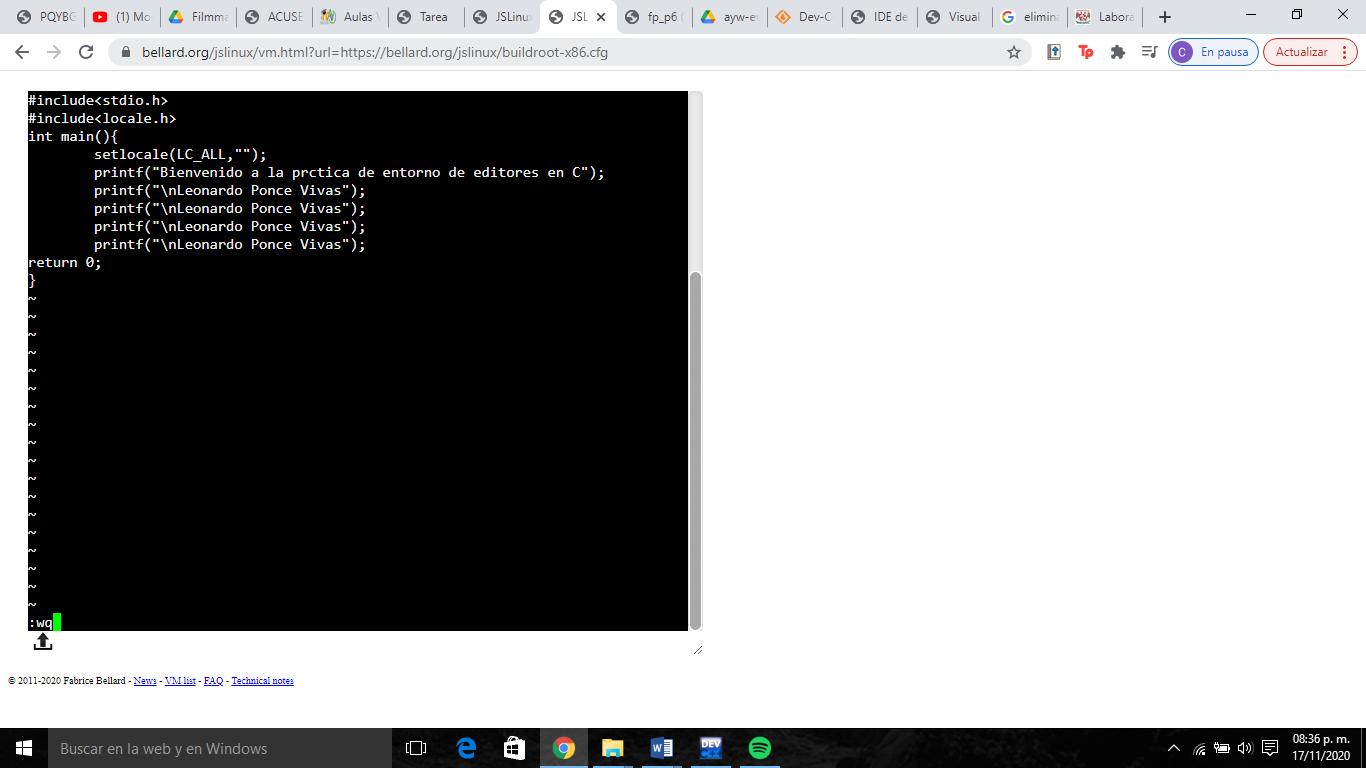


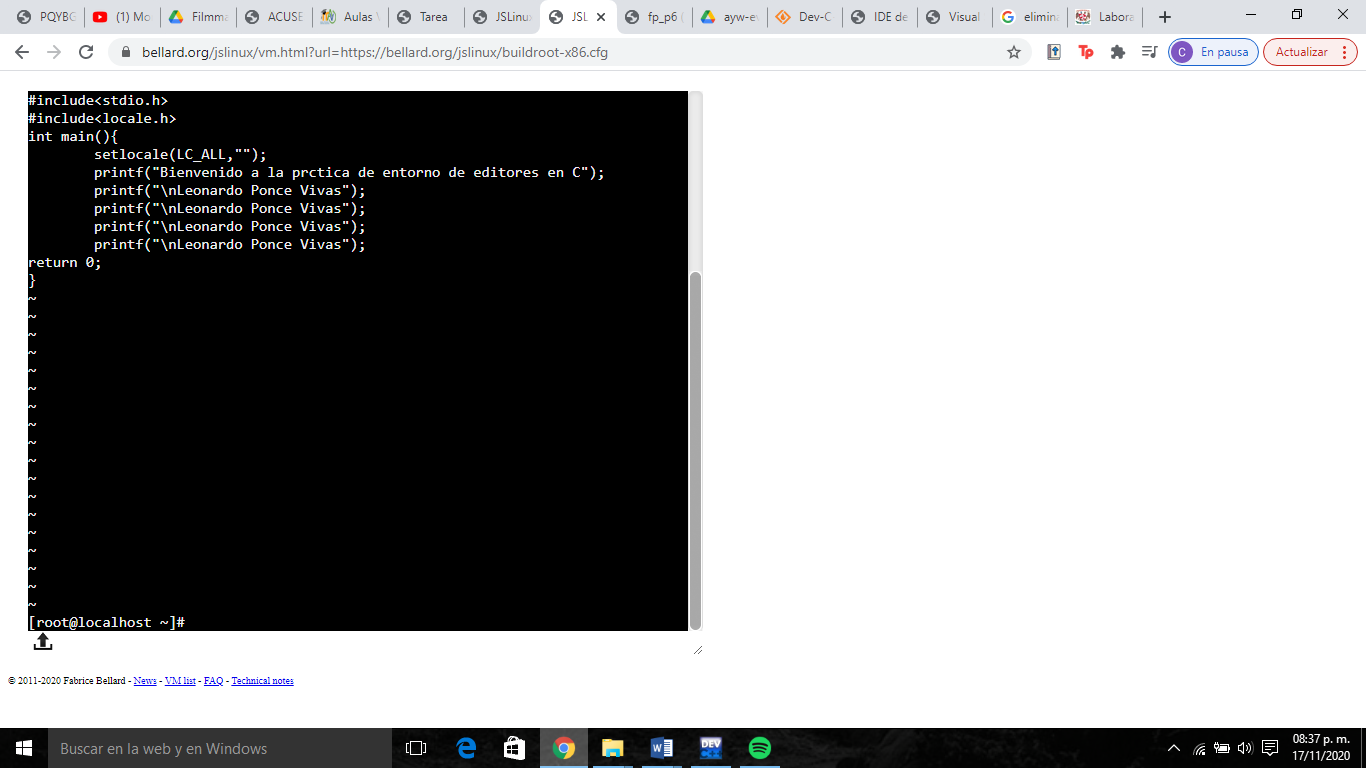
Se busca la palabra “Ponce” - usando “?”





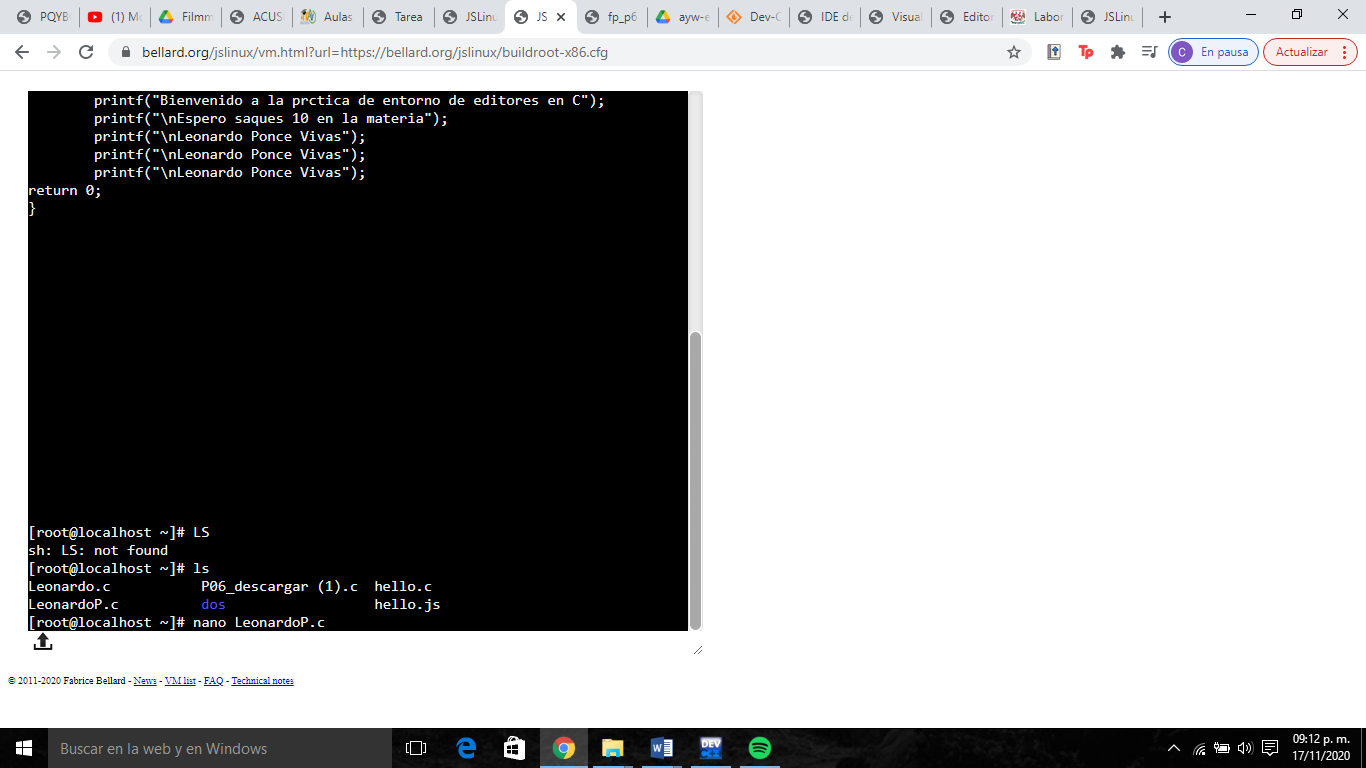
Se creó otro archivo; guardar como - usando “w”



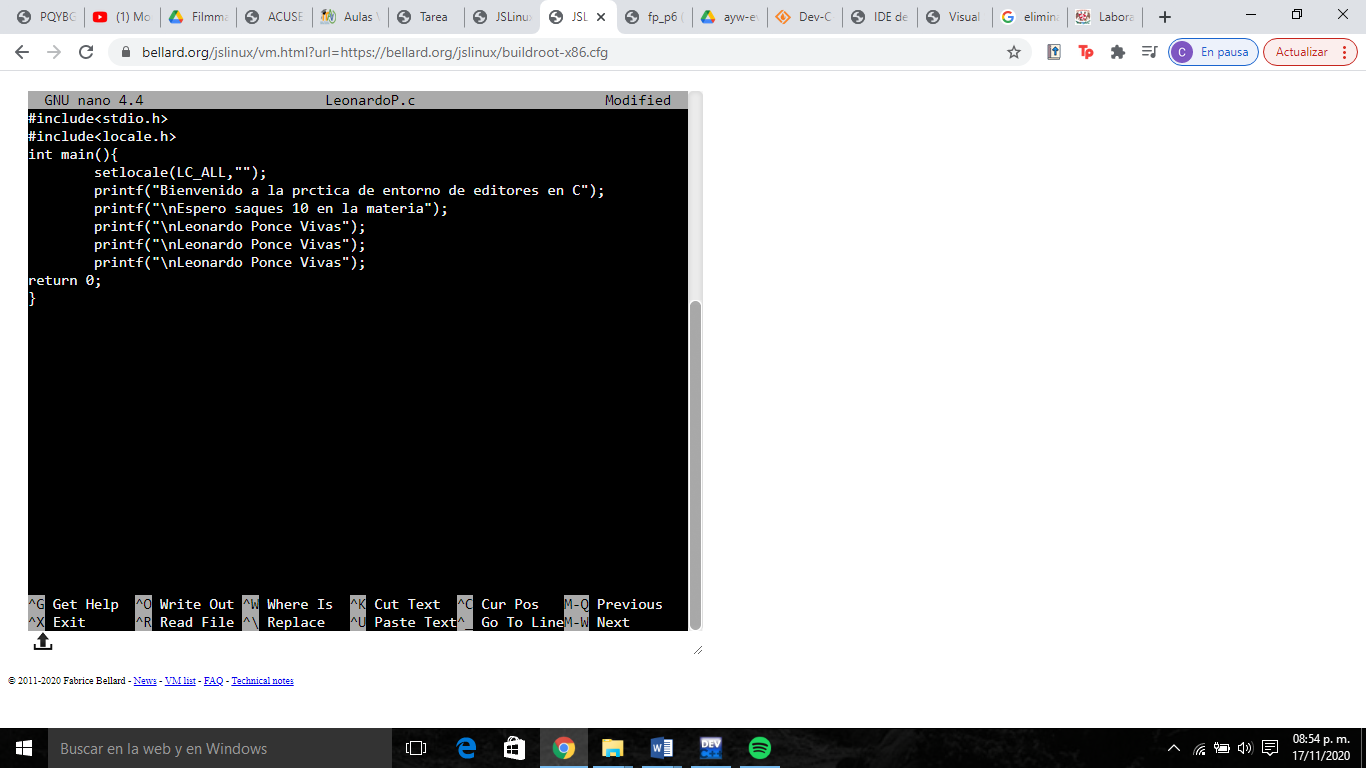


Se guardó y se salió del archivo - usando “wq”

Usar GNU NANO

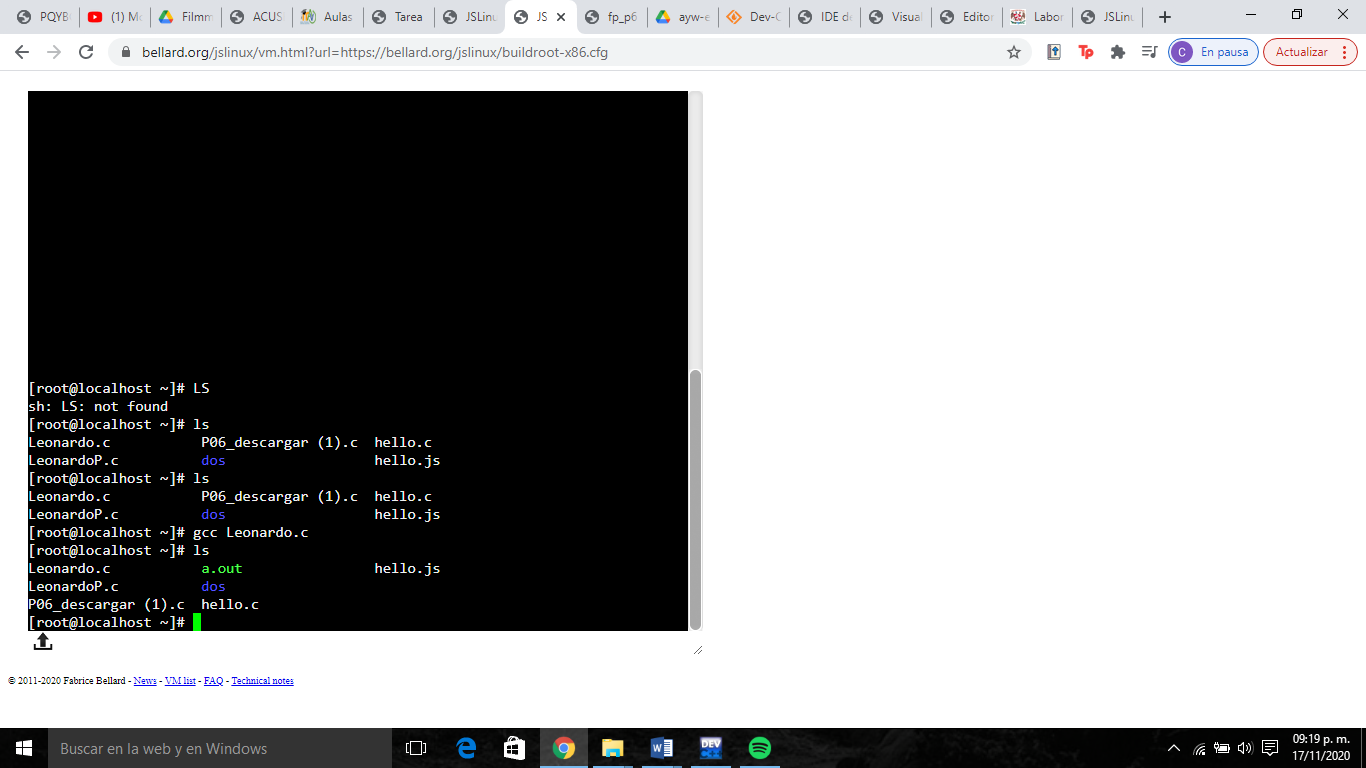


Entrar a nano

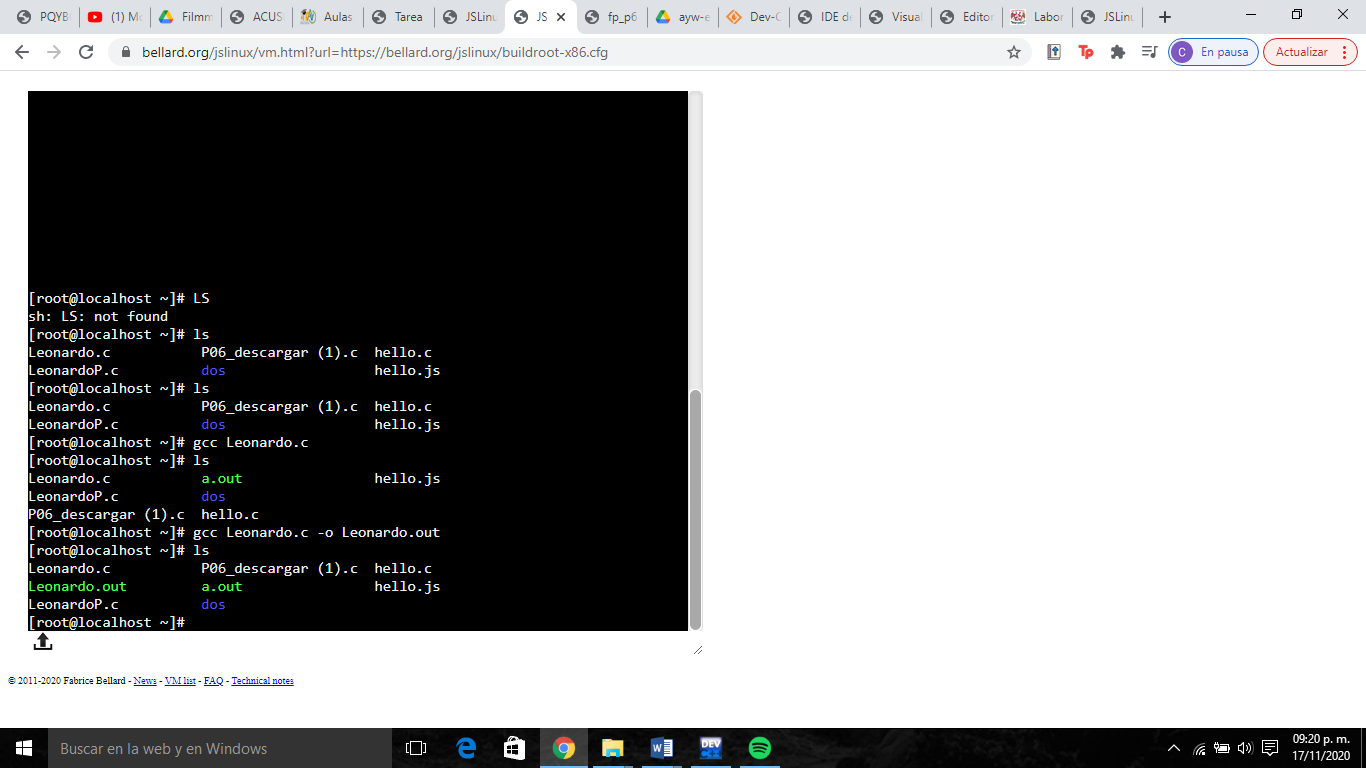


Modificar el archivo

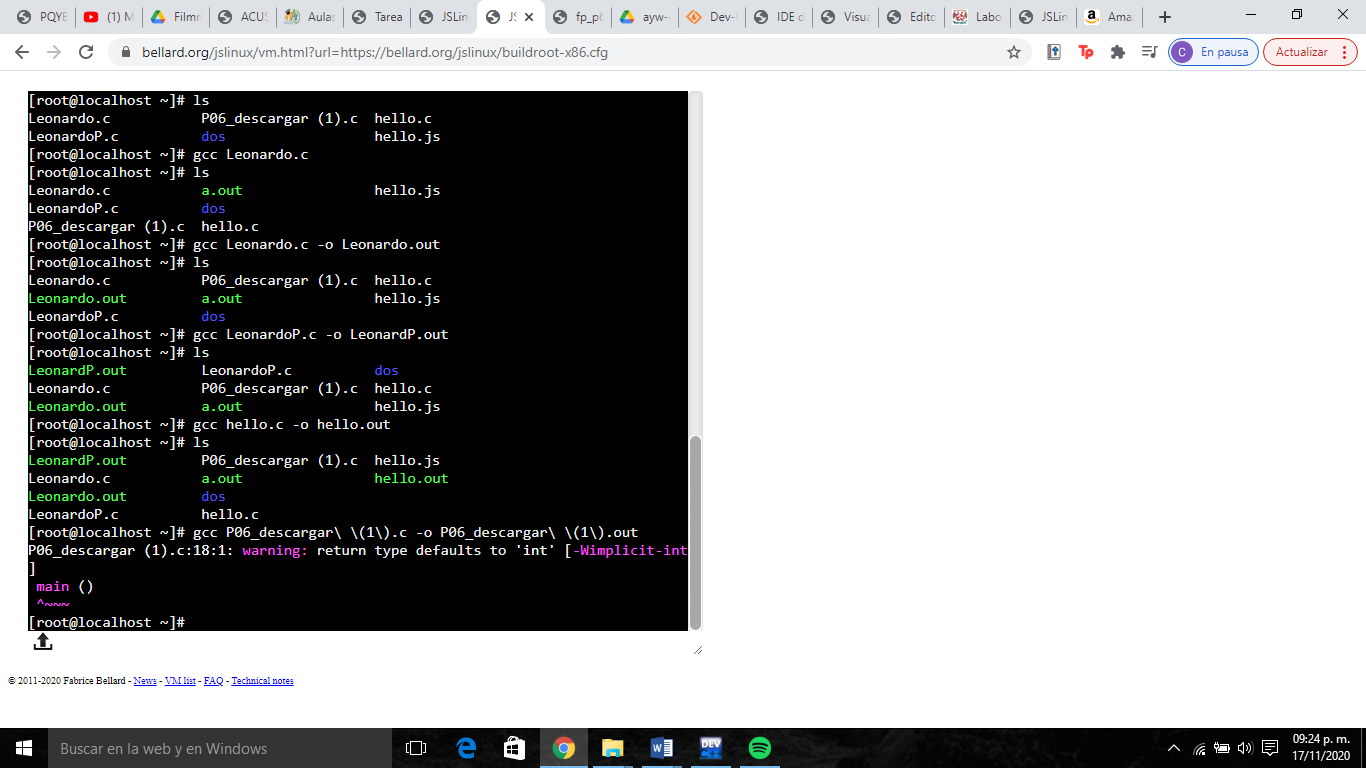
*Complilar*



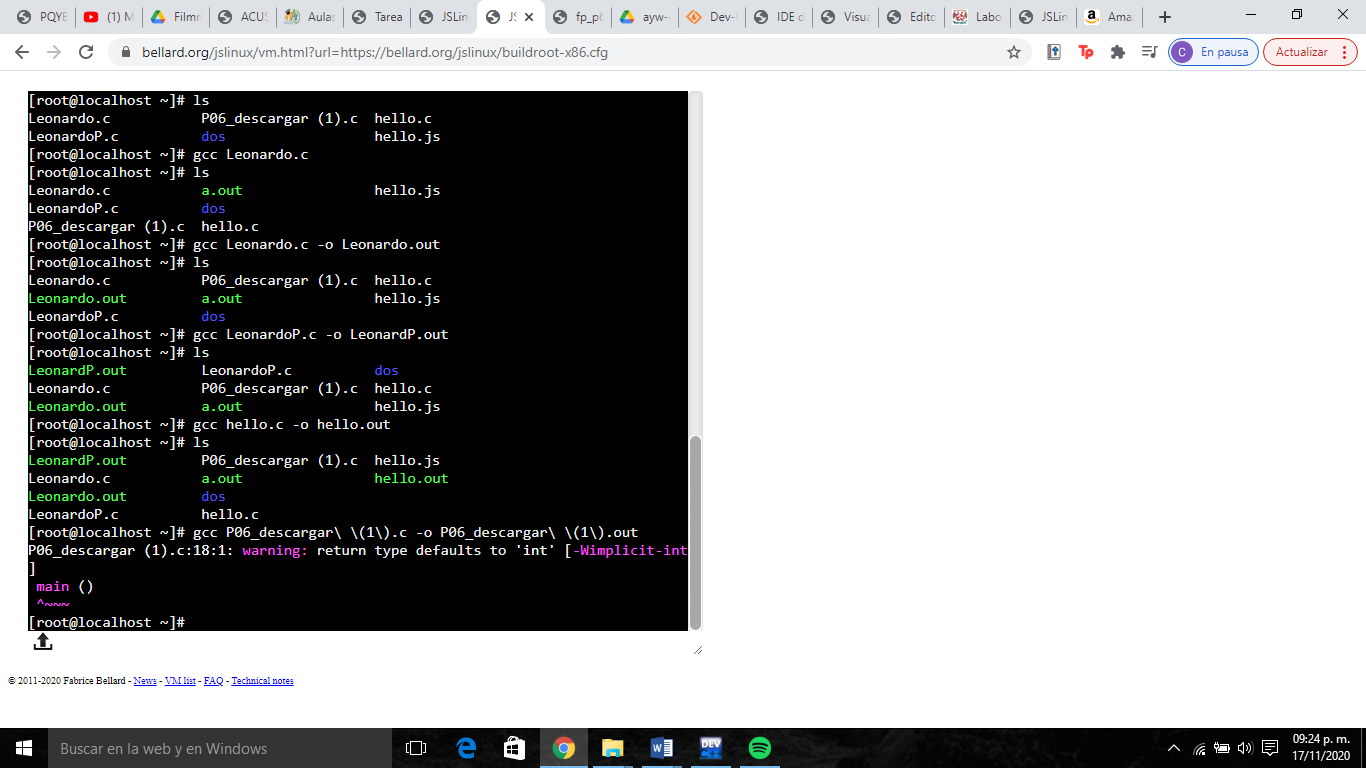
Usamos “gcc” con el nombre del archivo que queremos compilar



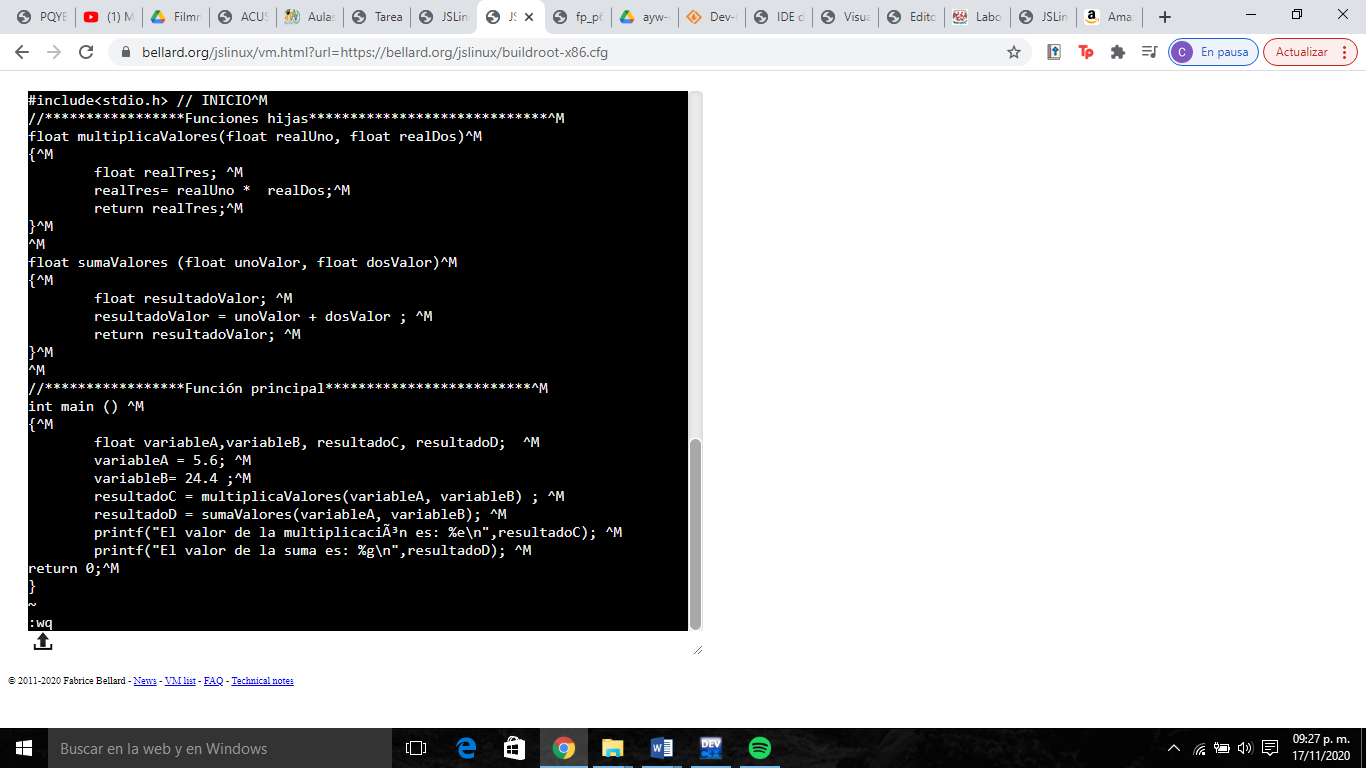
Se puso el nombre de salida



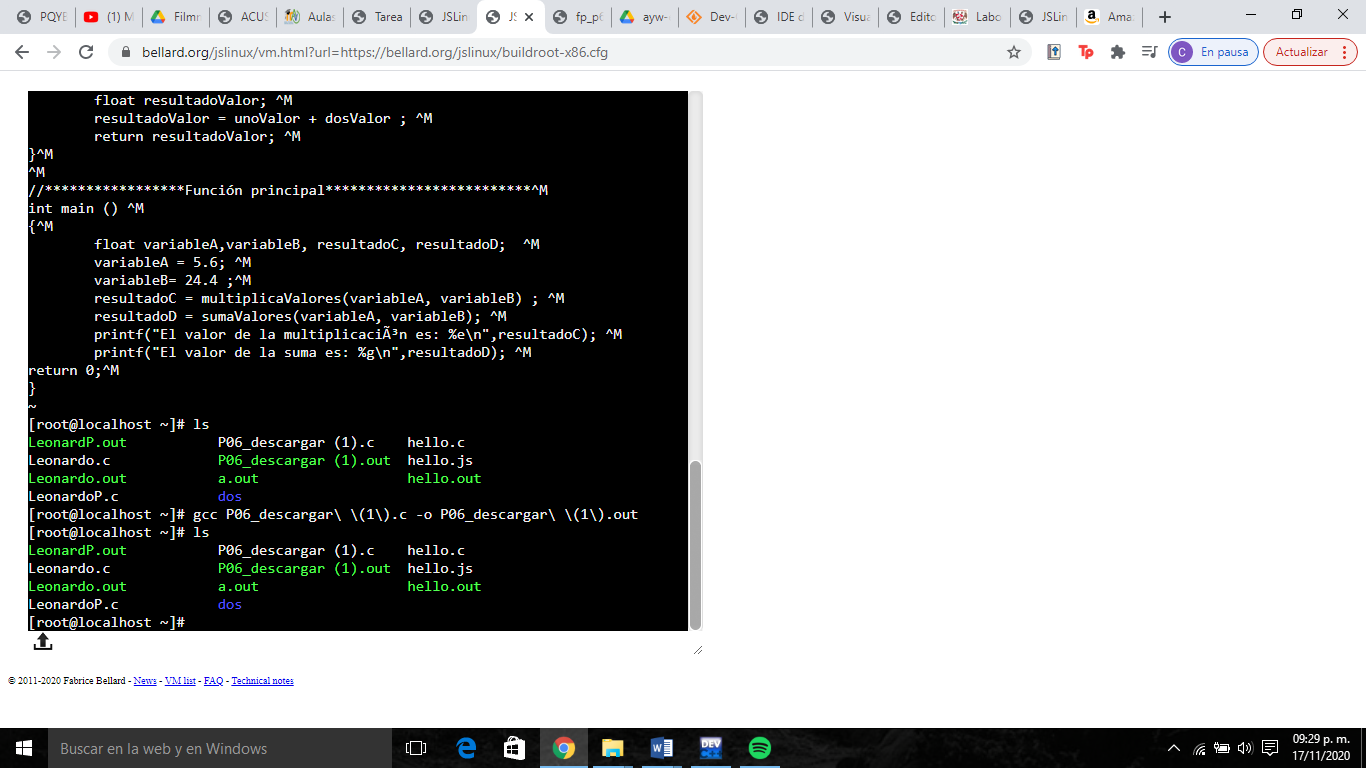
Compilar los demás



Se nos muestra una Advertencia debido a una falla en el código, en “main”



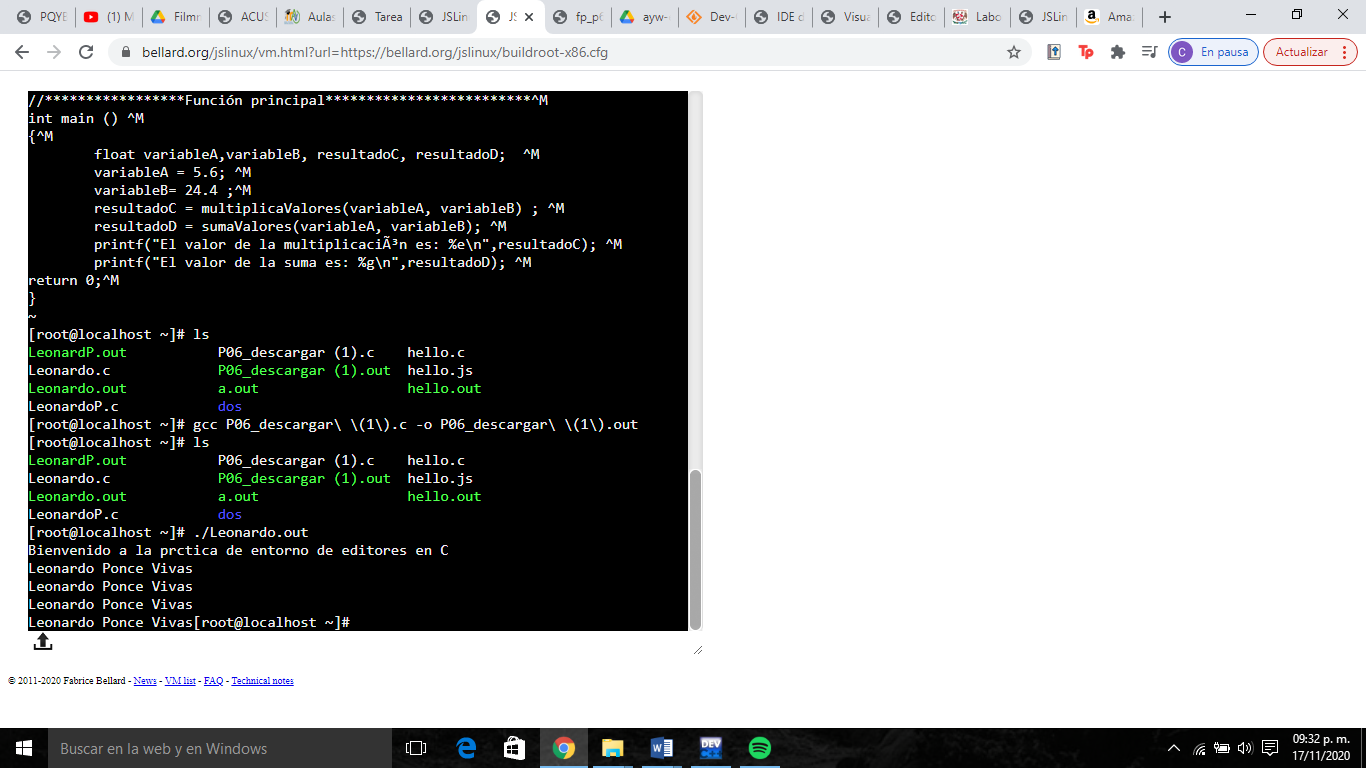
Solucionar error en “main”



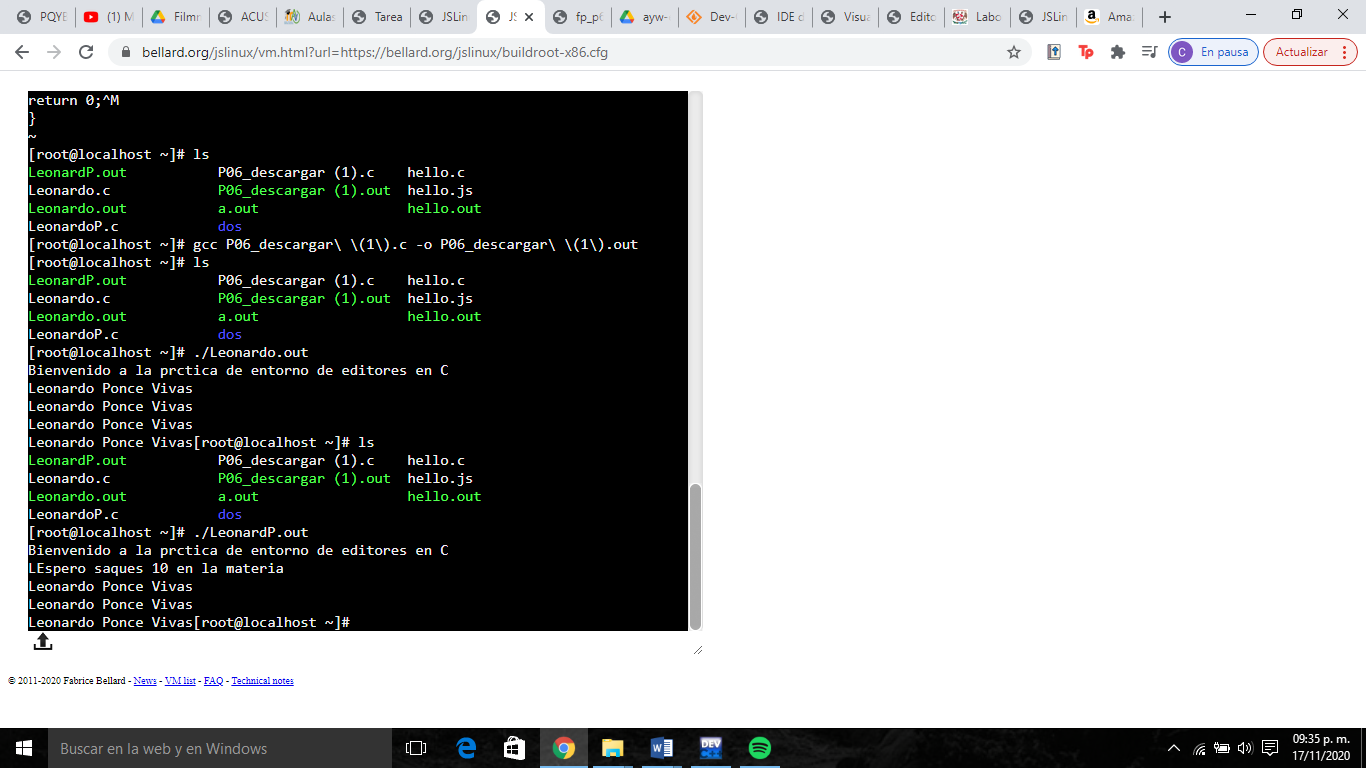
Compilar el archivo

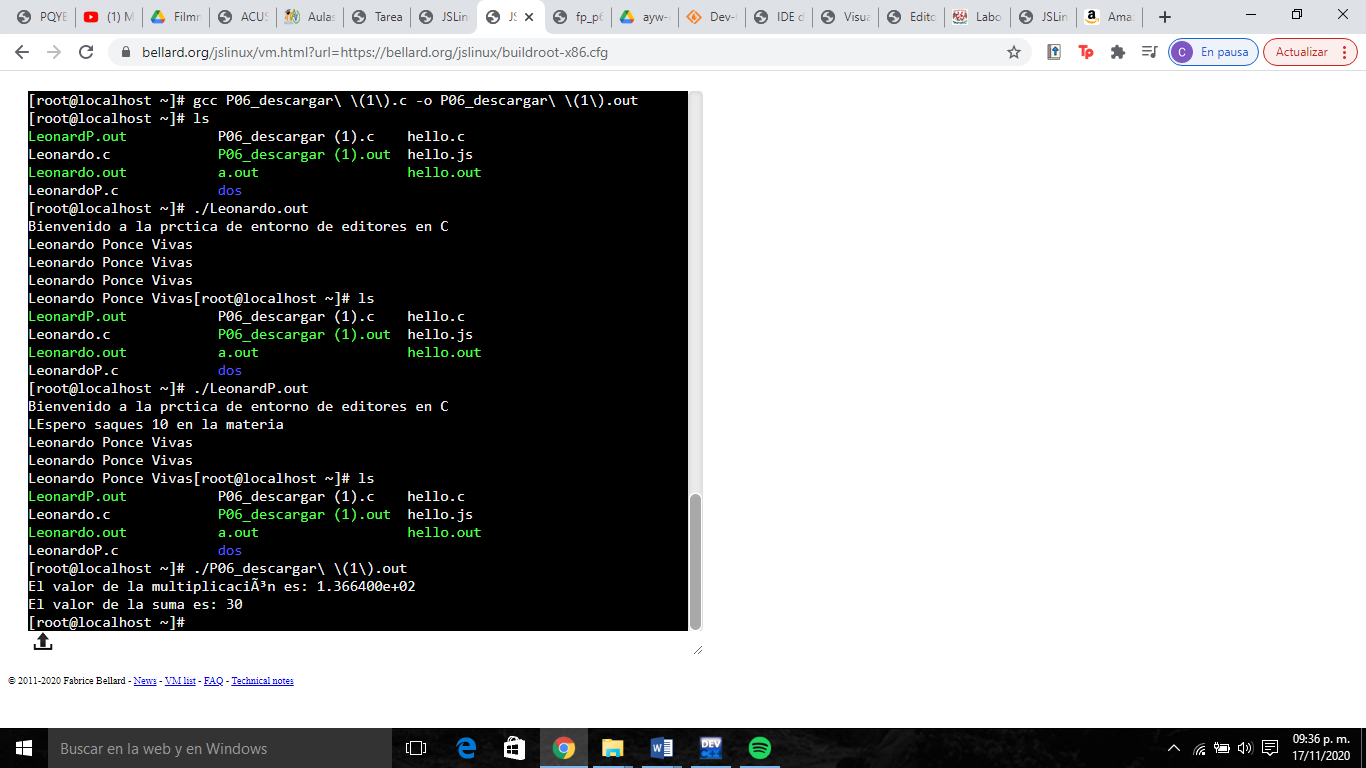
Tener todos los archivos compilados

*Ejecutar*



Ejecutamos con “./” con el archivo que queremos ejecutar





Hacer lo mismo con los demás

**Tarea:**

1.- Comparación entre Editor de Texto y Procesador de Texto

|  |  |
| --- | --- |
| Editor de texto | Procesador de texto |
| No cuenta con un formato | Existe la opción de un formato para el texto |
| Su peso es bajo | Su peso es alto |
| Permite crear y modificar archivos digitales compuestos únicamente por texto sin formato | Se pueden intercalar las palabras, agregar efectos artísticos, cambiar el tamaño de la letra |
| Programas:  Bloc de notas de Windows  [Visual Studio Code](https://code.visualstudio.com/)  [Geany](https://www.geany.org/)  [Notepad ++](https://notepad-plus-plus.org/download/v7.3.3.html)  Sublime Text | Programas:  WordPad  Microsoft Word  [LibreOffice Writer](https://es.libreoffice.org/)  [AbiWord](https://www.abisource.com/) |

2.- Indica los comandos ejecutados para compilar y para ejecutar un programa

**Compilar**

gcc “nombre del archvio”.c

Ejemplo:

gcc Leonardo.c

**Compilar con nombre de salida**

gcc “nombre del archvio”.c –o “nombre del archvio”.out

Ejemplo

gcc Leonardo.c –o Leonardo.out

**Compilar colocándolo en una biblioteca**

gcc “nombre del archvio”.c –o “nombre del archvio”.out -/nombre\_libreria

Ejemplo

gcc Leonardo.c –o Leonardo.out -/FundamentosP

**Ejecutar**

./”nombre del archivo”.out

Ejemplo

./Leonardo.out

**Ejecutar en Windows**

“nombre del archivo”.exe

Ejemplo

Leonardo.exe

**Ejecutar con argumentos existentes**

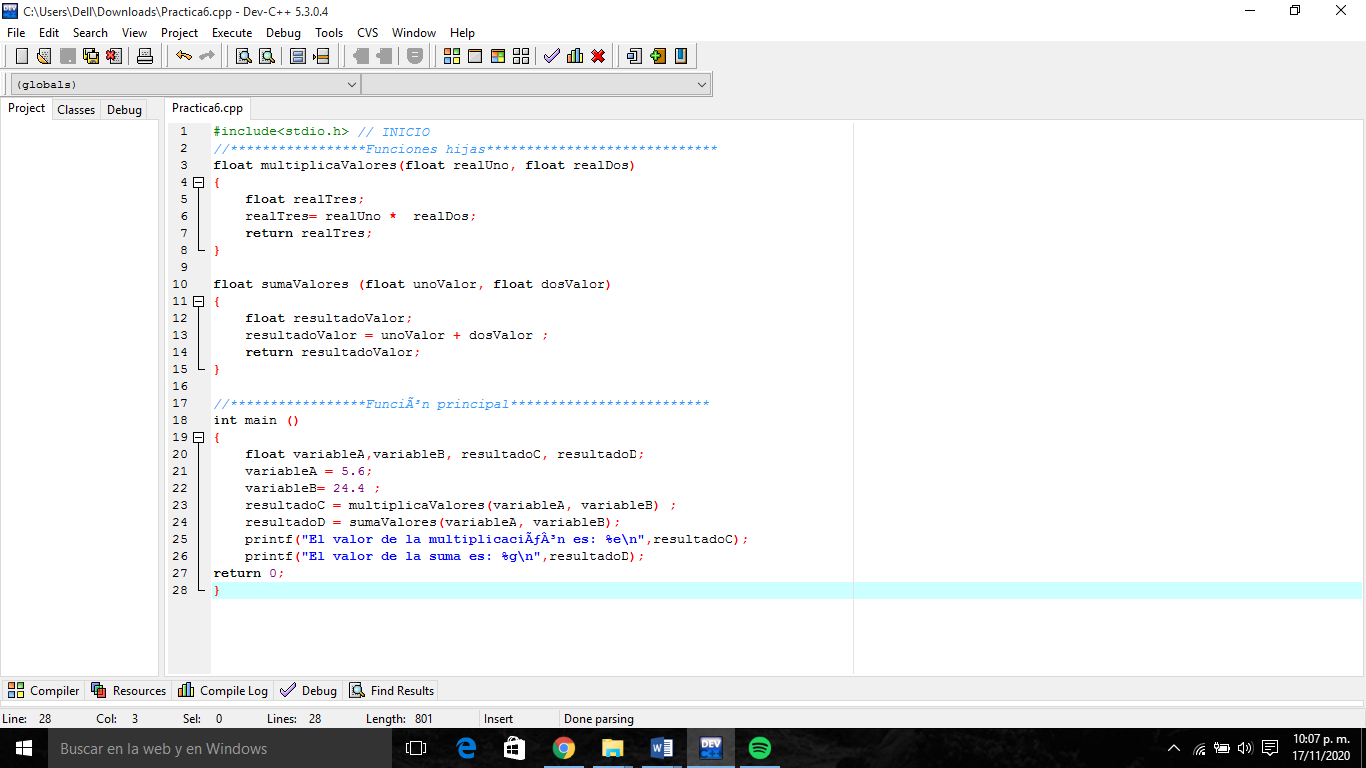
”nombre del archivo” “argumento1” “argumento2”

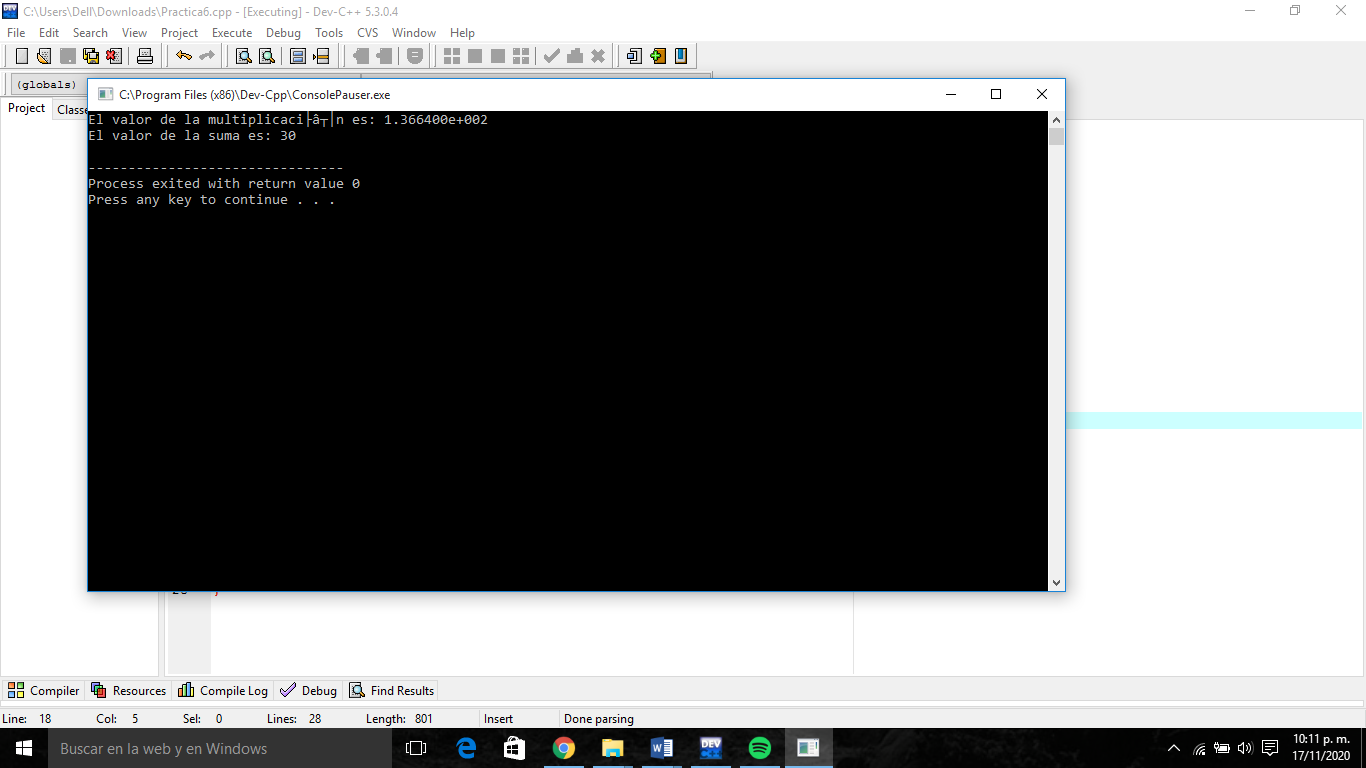
Ejemplo

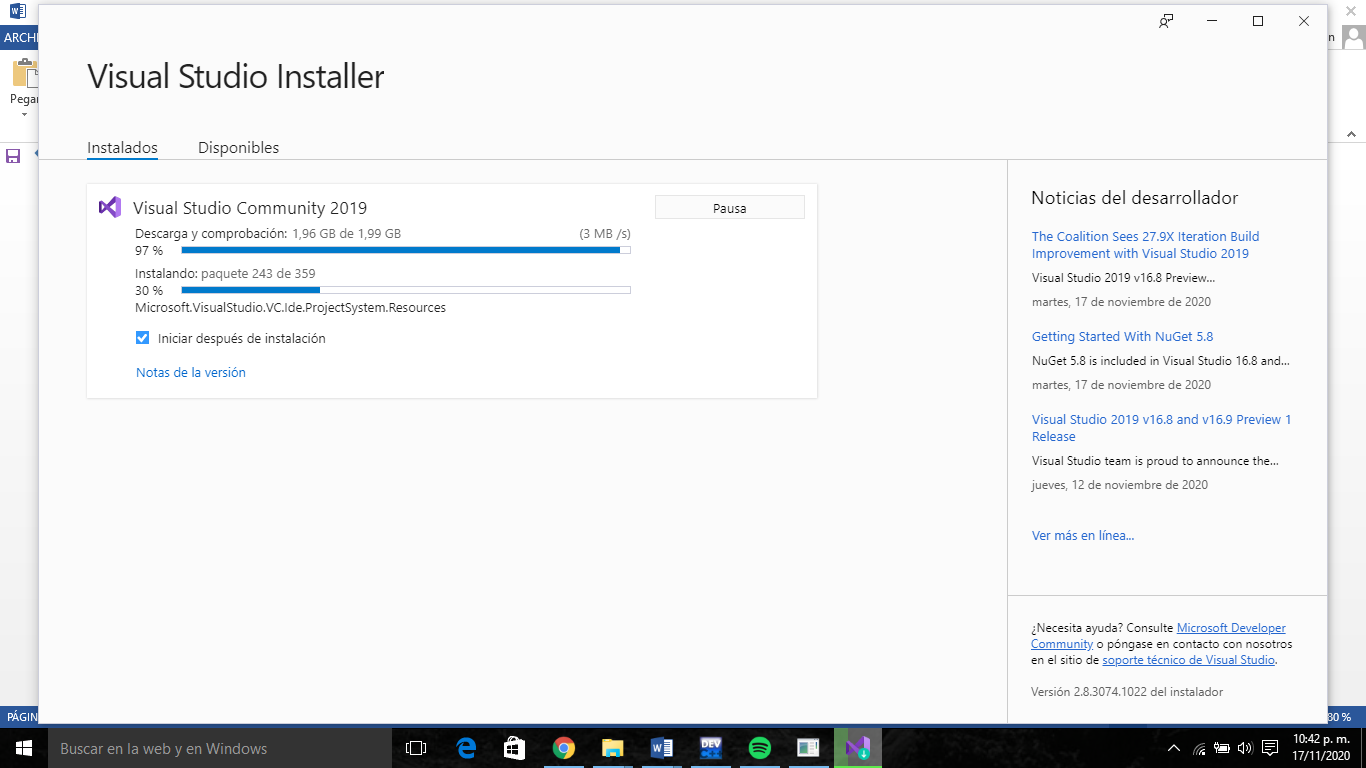
Leonardo argumento1 argumento2

3.- Compilación y prueba del programa antes mostrado en DEV C++ u otro IDE en sistema operativo Windows.

**DEV C++**





**IDE** 

Tuve problemas con la instalación de la aplicación, debido a la tardanza

No pude llevarlo acabo):

**Conclusión:**

En la práctica vimos un paso más sobre el desarrollo y práctica del programa Linux, usando el editor de texto para la creación de archivos de texto ya sea con grandes o pequeñas funciones. Haciendo usos de los atajos o controles para la elaboración de archivos, usándolos en varias formas como NANO, también lo practicamos en otros programas además de Linux como DEV C++, de esta manera se tiene un uso más amplio de la implementación de códigos. Todo esto se complementa con nuestros conocimientos en los algoritmos, llevándolos a otro nivel usando códigos, de tal manera que tenemos aún más herramientas para la solución de herramientas.

**Bibliografía:**

* Dr. Pedro Alberto Enríquez Palma. Editor VI. Consulta: septiembre de 2016. Disponible en: http://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf
* Francisconi Hugo Adrian. Nano. Consulta: septiembre de 2016. Disponible en: http://francisconi.org/linux/comandos/nano
* G2 Crowd. ATOM vs. Notepad++. Consulta: septiembre de 2016. Disponible en: https://www.g2crowd.com/compare/atom-vs-notepad
* Gerald Pfeifer (GCC team). GCC, the GNU Compiler Collection. Consulta: septiembre de 2015. Disponible en: https://gcc.gnu.org
* MinGW.org. MinGW - Minimalist GNU for Windows. Consulta: septiembre de 2015. Disponible en: http://www.mingw.org
* White-Hat Hacking. Uso de gcc bajo Linux. Consulta: septiembre de 2016. Disponible en: https://whitehathacking.wordpress.com/2010/10/31/uso-de-gcc-bajo-linux/
* Willus.org. Win32/64 C/C++ Compilers Page. Consulta: septiembre de 2016. Disponible en: http://www.willus.com/ccomp.shtml  Fabrice Bellard. Tiny C Compiler. Consulta: septiembre de 2015. Disponible en: http://bellard.org/tcc/
* Laboratorio Salas A y B. (2015). Retrieved November 4, 2020, from Unam.mx website: <http://lcp02.fi-b.unam.mx/>