	Pseudocódigo	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación
salas A y B

Profesor: Alejandro Pimentel

Asignatura: Fundamentos de Programación

Grupo: 135

No de Práctica(s): 6

Integrante(s): Lorena Basurto Amezcu

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 2858

Semestre: 2020-1

Fecha de entrega: Septiembre 30, 2019.

Observaciones: La última actividad no es correcta.
No muestras ninguna evidencia de haber compilado
y ejecutado tu programa correctamente.

CALIFICACIÓN: **8**

Objetivo:

Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

Desarrollo:

- Procesador de texto:

El **procesador de texto** es una aplicación informática para la creación, edición, modificación y procesamiento de documentos de texto con formato (tal como el tipo y tamaño de la tipografía, adición de gráficos, etc.), a diferencia de los editores de texto, que manejan solo texto simple.

Los procesadores de textos son una clase de software con múltiples funcionalidades para la redacción, con diferentes tipografías, tamaños de letras o caracteres, colores, tipos de párrafos, efectos artísticos y otras opciones.

- Editor de texto:

Un **editor de texto** es un programa informático que permite crear y modificar archivos digitales compuestos únicamente por textos sin formato, conocidos comúnmente como archivos de texto o “texto plano”. El programa lee el archivo e interpreta los bytes leídos según el código de caracteres que usa el editor.

- Compilador:

En informática, un **compilador** es un tipo de traductor que transforma un programa entero de un lenguaje de programación (llamado código fuente) a otro.

• Actividad: Texto plano

El texto plano es el tipo de archivo más sencillo que hay, ya que en su contenido no hay otra cosa más que el texto que lo conforma. Hacer una investigación muy somera acerca de los siguientes tipos de archivos:

- *txt*:

Un *archivo de texto simple*, *texto sencillo* o *texto sin formato* (también llamado *texto llano* o *texto simple*; en inglés «*plain text*»), es un archivo informático que contiene únicamente texto formado solo por caracteres que son legibles por humanos, careciendo de cualquier tipo de formato tipográfico.

- *Markdown*:

Markdown es un *lenguaje de marcado* que facilita la aplicación de formato a un texto empleando una serie de caracteres de una forma especial. En principio, fue pensado para elaborar textos cuyo destino iba a ser la web con más rapidez y sencillez que si se empleara directamente HTML. Y si bien ese suele ser el mejor



uso que se le puede dar, también se puede emplear para cualquier tipo de texto, independientemente de cuál vaya a ser su destino.

- *Html:*
HTML, siglas en inglés de *HyperText Markup Language* ('lenguaje de marcas de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.



- *LaTeX:*
Es un *sistema de composición de textos*, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.
- *Csv:*
La extensión de archivo CSV significa Comma Separated Values (Valores separados por comas). El formato es utilizado en muchos programas de bases de datos, hojas de cálculo y gestores de contactos para almacenar listas de información. Como un archivo de texto, el formato es ampliamente compatible.

• Editores

Un editor de texto es un programa que es capaz de editar texto plano. No confundir con los procesadores de texto.

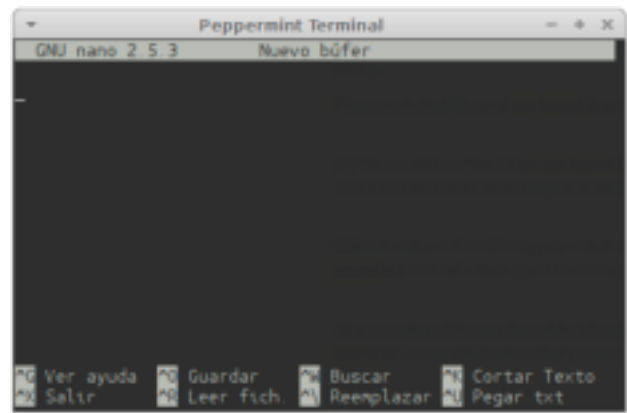
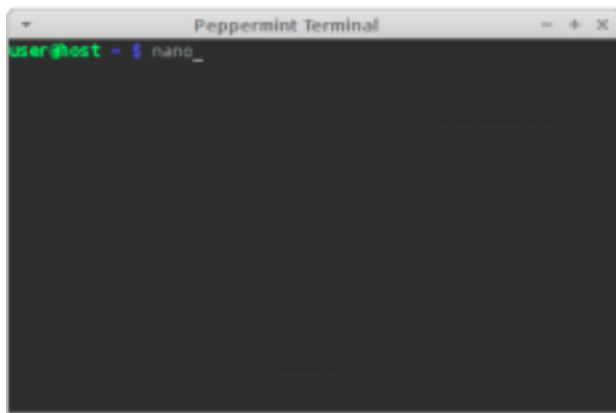
- | | | |
|------------|-------------|----------------|
| ○ Atom | ○ Geany | ○ Pico |
| ○ Bluefish | ○ Emacs | ○ Sublime Text |
| ○ Brackets | ○ Nano | ○ Vim |
| ○ Gedit | ○ Notepad++ | |

En general, los *editores* difieren en su modo de uso y en las características que ofrecen.

- | | |
|---|---------------------------------------|
| ○ Resaltado de palabras clave | ○ Integración de compilador |
| ○ Autocompletado | ○ Integración de control de versiones |
| ○ Lista de elementos definidos | ○ Integración de terminal |
| ○ Autosangrado | ○ Búsquedas avanzadas |
| ○ Identificación de pares de paréntesis | |

• Nano

Un editor de texto de terminal. Muy simple y básico, útil principalmente para editar textos cortos como los commits de git.



```
$ git config --global core.editor nano
```

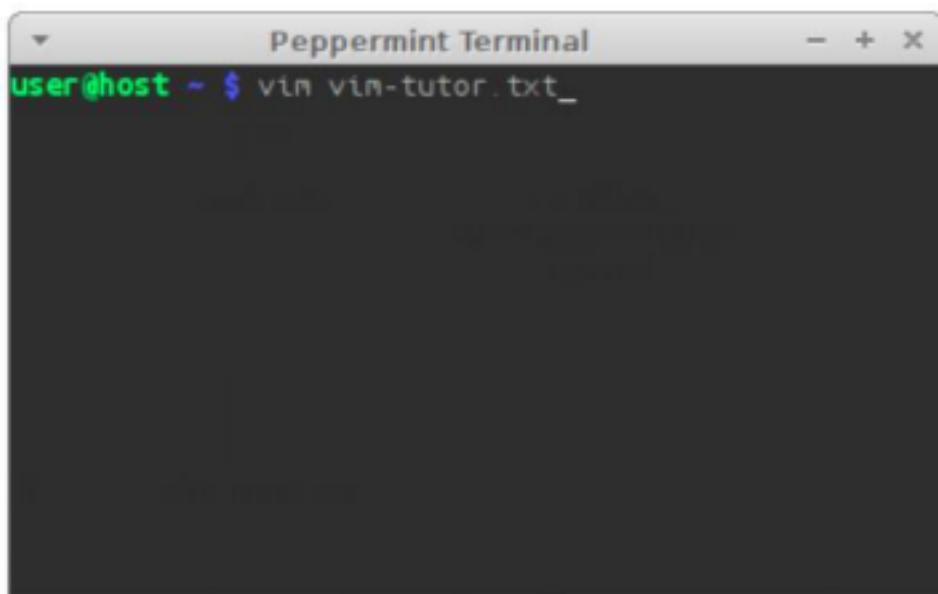
```
$ git config --global core.editor vi
```

- Vi

Otro editor de terminal, aunque también tiene versiones gráficas. Este es uno de los editores más populares, con más características, y que los expertos aseguran que ofrecen mayor ganancia de velocidad para editar textos. La razón es que este editor es bastante diferente a lo que están acostumbrados. Se requiere más tiempo para aprenderlo a usar y para dominarlo.

- Actividad: Vi

Seguir el tutor de Vim.




```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda
[Icons]
-----
Lección 2.1:  MANDATOS PARA BORRAR

** Escriba dw para borrar hasta el final de una palabra **

1. Pulse <ESC> para asegurarse de que está en el modo Normal.
2. Mueva el cursor a la línea de abajo señalada con --->.
3. Mueva el cursor al comienzo de una palabra que desee borrar.
4. Pulse  dw  para hacer que la palabra desaparezca.

NOTA: Las letras  dw  aparecerán en la última línea de la pantalla cuando
      las escriba. Si escribe algo equivocado pulse <ESC> y comience de nuevo.

---> Hay algunas palabras que no pertenecen a esta frase.

-----
154,0-1      18%
```

```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda
[Icons]
-----
Lección 2.2:  MÁS MANDATOS PARA BORRAR

** Escriba d$ para borrar hasta el final de la línea. **

1. Pulse <ESC> para asegurarse de que está en el modo Normal.
2. Mueva el cursor a la línea de abajo señalada con --->.
3. Mueva el cursor al final de la línea correcta (DESPUÉS del primer . ).
4. Escriba d$ para borrar hasta el final de la línea.

---> Alguien ha escrito el final de esta línea dos veces.

-----
180,0-1      21%
```

```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda

-----
Lección 2.4: UNA EXCEPCIÓN AL 'MANDATO-OBJETO'

** Escriba dd para borrar una línea entera. **

Debido a la frecuencia con que se borran líneas enteras, los diseñadores
de Vim decidieron que sería más fácil el escribir simplemente dos des en
una fila para borrar una línea.

1. Mueva el cursor a la segunda línea de la lista de abajo.
2. Escriba dd para borrar la línea.
3. Muévase ahora a la cuarta línea.
4. Escriba 2dd (recuerde número-mandato-objeto) para borrar las dos
líneas.

1) Las rosas son rojas,
3) El cielo es azul,
6) El azúcar es dulce,
7) Y así- eres tu.

-----
222,7 27%
```

```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda

-----
Lección 2.5: EL MANDATO DESHACER

** Pulse u para deshacer los últimos mandatos,
U para deshacer una línea entera. **

1. Mueva el cursor a la línea de abajo señalada con ---> y sitúelo bajo el
primer error.
2. Pulse x para borrar el primer carácter erróneo.
3. Pulse ahora u para deshacer el último mandato ejecutado.
4. Ahora corrija todos los errores de la línea usando el mandato x.
5. Pulse ahora U mayúscula para devolver la línea a su estado original.
6. Pulse ahora u unas pocas veces para deshacer lo hecho por U y los
mandatos previos.
7. Ahora pulse CTRL-R (mantenga pulsada la tecla CTRL y pulse R) unas
pocas veces para volver a ejecutar los mandatos (deshacer lo deshecho).
---> Corrija los errores de esta línea y vuelva a ponerlos con deshacer.
8. Estos mandatos son muy útiles. Ahora pase al resumen de la Lección 2.

-----
244,8-1 30%
```

```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda

-----
Lección 2.5: EL MANDATO DESHACER

** Pulse u para deshacer los últimos mandatos,
   U para deshacer una línea entera. **

1. Mueva el cursor a la línea de abajo señalada con ---> y sitúelo bajo el
   primer error.
2. Pulse x para borrar el primer carácter erróneo.
3. Pulse ahora u para deshacer el último mandato ejecutado.
4. Ahora corrija todos los errores de la línea usando el mandato x.
5. Pulse ahora U mayúscula para devolver la línea a su estado original.
6. Pulse ahora u unas pocas veces para deshacer lo hecho por U y los
   mandatos previos.
7. Ahora pulse CTRL-R (mantenga pulsada la tecla CTRL y pulse R) unas
   pocas veces para volver a ejecutar los mandatos (deshacer lo deshecho).
---> Corrija los errores de esta línea y vuelva a ponerlos con deshacer.
8. Estos mandatos son muy útiles. Ahora pase al resumen de la Lección 2.

-----
```

```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda

-----
Lección 3.1: EL MANDATO «PUT» (poner)

** Pulse p para poner lo último que ha borrado después del cursor. **

1. Mueva el cursor al final de la lista de abajo.
2. Escriba dd para borrar la línea y almacenarla en el buffer de Vin.
3. Mueva el cursor a la línea que debe quedar por debajo de la
   línea a mover.
4. Estando en mod Normal, pulse p para restituir la línea borrada.
5. Repita los pasos 2 a 4 para poner todas las líneas en el orden correcto.

a) Las rosas son rojas,
b) Las violetas son azules,
c) La inteligencia se aprende,
d) ¿Puedes aprenderla tu?

-----

289,6 36%
```



```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda

-----

Lección 3.2: EL MANDATO «REPLACE» (reemplazar)

** Pulse r y un carácter para sustituir el carácter sobre el cursor. **

1. Mueva el cursor a la primera línea de abajo señalada con --->.
2. Mueva el cursor para situarlo bajo el primer error.
3. Pulse r y el carácter que debe sustituir al erróneo.
4. Repita los pasos 2 y 3 hasta que la primera línea esté corregida.
---> ¿Cuándo esta línea fue escrita alguien pulsó algunas teclas equivocadas?
---> ¿Cuándo esta línea fue escrita alguien pulsó algunas teclas equivocadas?

312,53 39%
```

```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo  Editar  Herramientas  Sintaxis  Buffers  Ventana  Ayuda

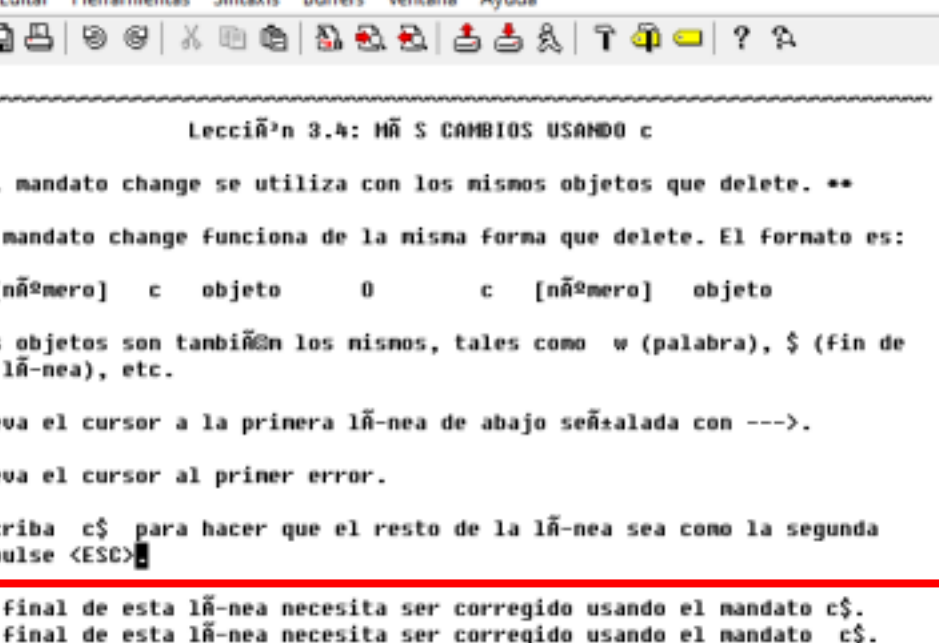
-----

Lección 3.3: EL MANDATO «CHANGE» (cambiar)

** Para cambiar parte de una palabra o toda ella escriba cw . **

1. Mueva el cursor a la primera línea de abajo señalada con --->.
2. Sitúe el cursor en la u de lubrs.
3. Escriba cw y corrija la palabra (en este caso, escriba 'á-nea').
4. Pulse <ESC> y mueva el cursor al error siguiente (el primer carácter
   que deba cambiarse).
5. Repita los pasos 3 y 4 hasta que la primera frase sea igual a la segunda.
---> Esta línea tiene unas pocas palabras que corregir usando el mandato change.
---> Esta línea tiene unas pocas palabras que corregir usando el mandato change.

337,0-1 42%
```



```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo Editar Herramientas Sintaxis Buffers Ventana Ayuda
[Icons]
-----
Lección 3.4: Hacer cambios usando c

** El mandato change se utiliza con los mismos objetos que delete. **

1. El mandato change funciona de la misma forma que delete. El formato es:

    [número] c objeto 0 c [número] objeto

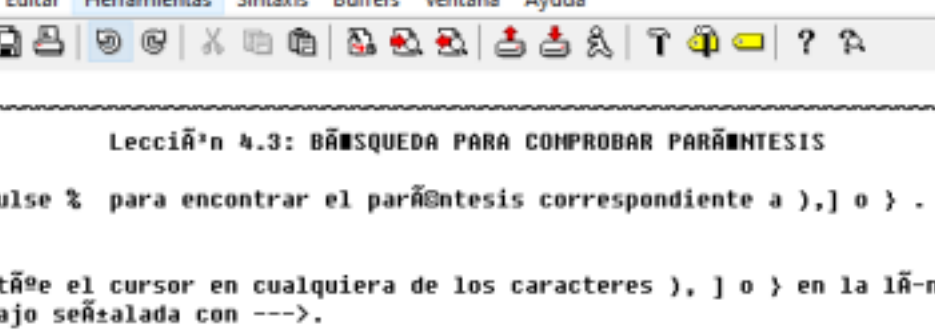
2. Los objetos son también los mismos, tales como w (palabra), $ (fin de
   la línea), etc.

3. Mueva el cursor a la primera línea de abajo señalada con --->.

4. Mueva el cursor al primer error.

5. Escriba c$ para hacer que el resto de la línea sea como la segunda
   y pulse <ESC>.

---> El final de esta línea necesita ser corregido usando el mandato c$.
---> El final de esta línea necesita ser corregido usando el mandato c$.
-----
359,19 45%
```



vim-tutor.txt (~\Downloads) - GVIM

Archivo Editar Herramientas Sintaxis Buffers Ventana Ayuda

Lección 4.3: BÚSQUEDA PARA COMPROBAR PARÉNTESIS

**** Pulse % para encontrar el paréntesis correspondiente a),] o } . ****

1. Sitúe el cursor en cualquiera de los caracteres),] o } en la línea de abajo señalada con --->.
2. Pulse ahora el carácter % .
3. El cursor deberá situarse en el paréntesis (, corchete [o llave { correspondiente.
4. Pulse % para mover de nuevo el cursor al paréntesis, corchete o llave correspondiente.

---> Esto (es una línea de prueba con (, [,], {, y } en ella.).

Nota: Esto es muy útil en la detección de errores en un programa con paréntesis, corchetes o llaves desaparejos.

447,66 57%

```
vim-tutor.txt + (~\Downloads) - GVIM
Archivo Editar Herramientas Sintaxis Buffers Ventana Ayuda
[Icons]

-----
Lección 4.4: UNA FORMA DE CAMBIAR ERRORES

** Escriba :s/viejo/nuevo/g para sustituir 'viejo' por 'nuevo'. **

1. Mueva el cursor a la línea de abajo señalada con --->.

2. Escriba :s/laas/las/ <INTRO> . Tenga en cuenta que este mandato cambia
sólo la primera aparición en la línea de la expresión a cambiar.

---> Las mejores épocas para ver las flores son las primaveras.

3. Para cambiar todas las apariciones de una expresión entre dos líneas
escriba :#,#s/viejo/nuevo/g donde #,# son los números de las dos
líneas. Escriba :%s/viejo/nuevo/g para hacer los cambios en todo
el fichero.

-----
466,3 60%
```

```
vim-tutor.txt (~\Downloads) - GVIM
Archivo Editar Herramientas Sintaxis Buffers Ventana Ayuda
[Icons]

-----
Lección 5.2: MÁS SOBRE GUARDAR FICHEROS

** Para guardar los cambios hechos en un fichero,
escriba :w NOMBRE_DE_FICHERO. **

1. Escriba :!dir o :!ls para ver una lista de su directorio.
Ya sabe que debe pulsar <INTRO> después de ello.

2. Elija un nombre de fichero que todavía no exista, como TEST.

3. Ahora escriba :w TEST (donde TEST es el nombre de fichero elegido).

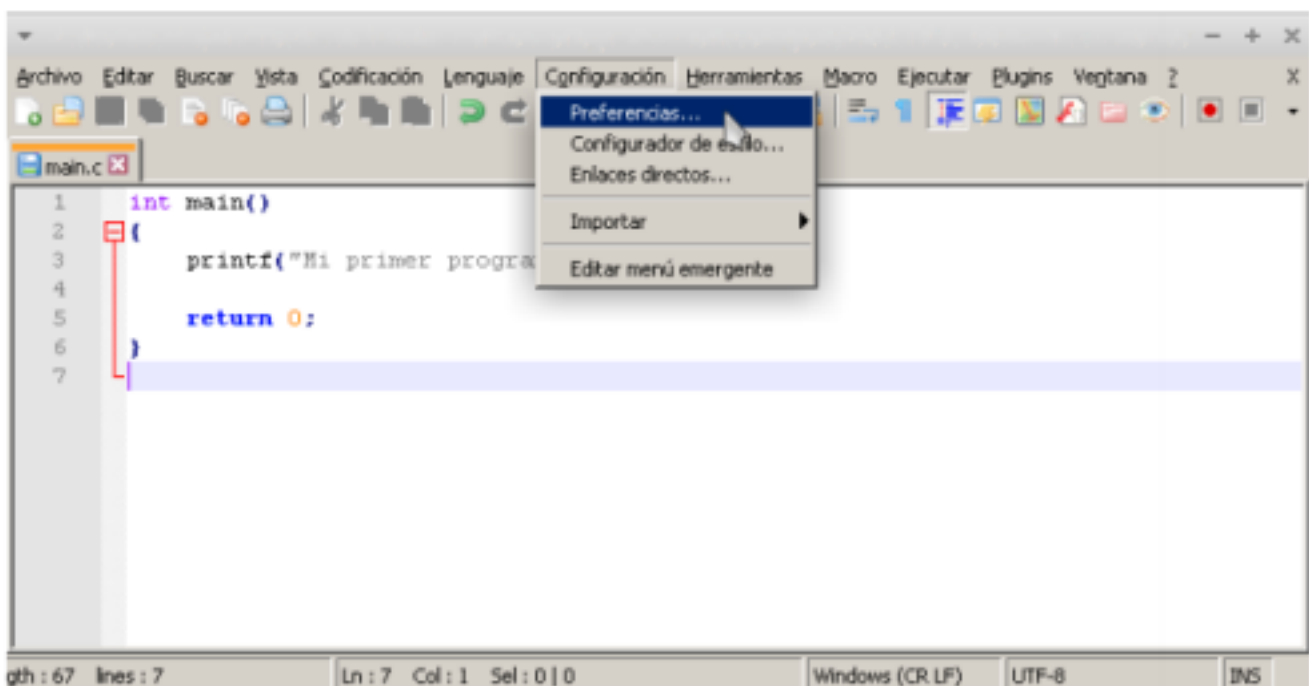
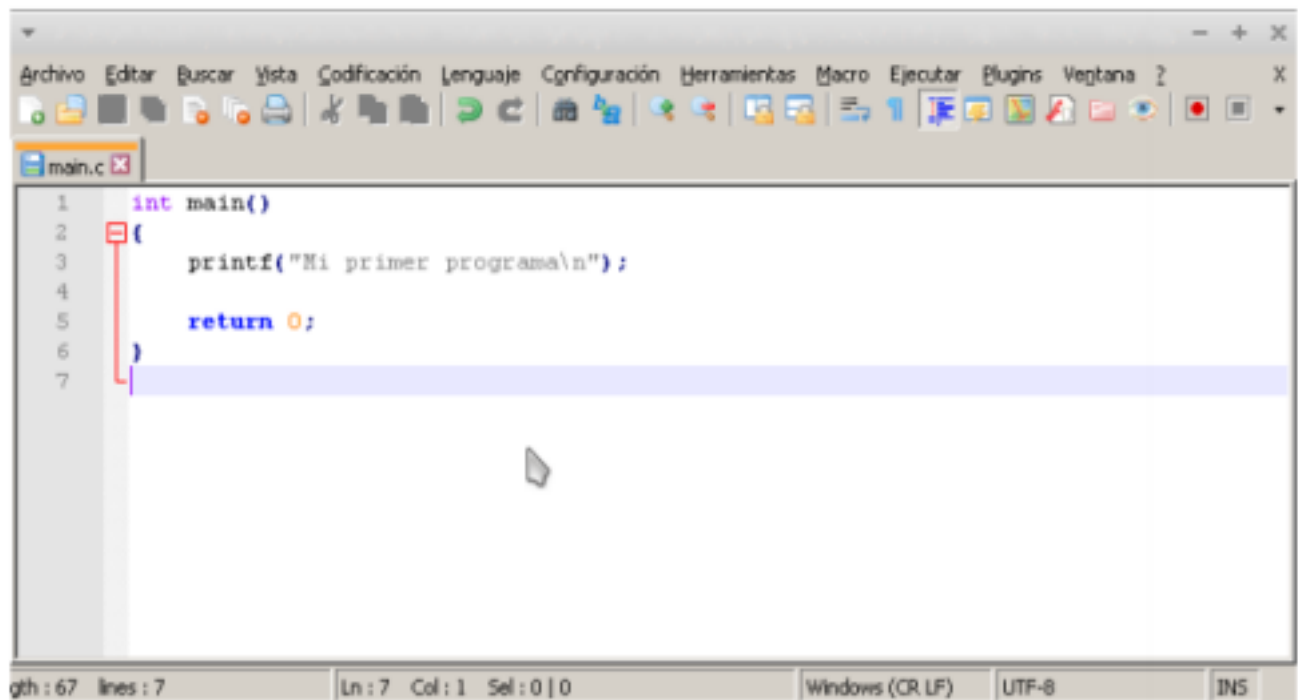
4. Esta acción guarda todo el fichero (Vim Tutor) bajo el nombre TEST.
Para comprobarlo escriba :!dir de nuevo y vea su directorio.

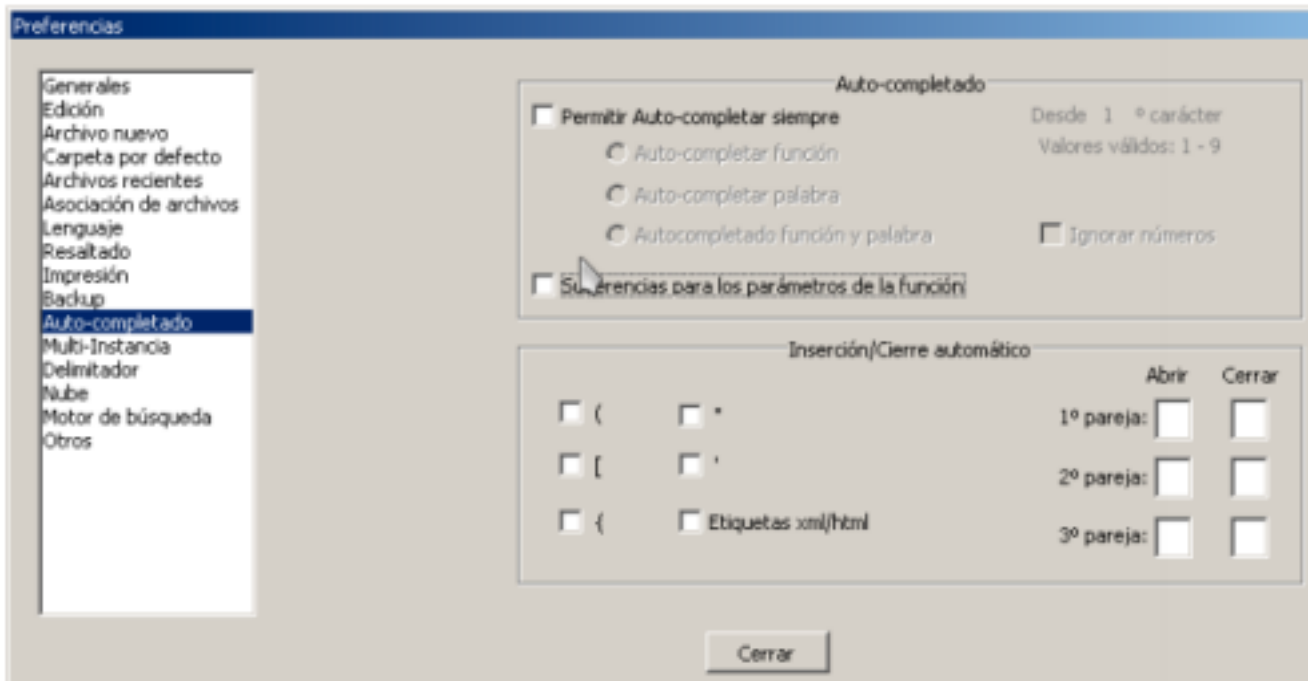
---> Tenga en cuenta que si sale de Vim y entra de nuevo con el nombre de
fichero TEST, el fichero será una copia exacta del tutor cuando lo
ha guardado.

-----
"TEST3" [Nuevo][UNIX] 769L, 28041C escritos 535,63 69%
```


- **Notepad ++**

Notepad++ es un editor gráfico mucho más parecido a lo que están acostumbrados.





- **Compilar**

```
Peppermint Terminal
user@host ~ $ gcc main.c -o main_
```

- **Correr**

```
Peppermint Terminal
user@host ~ $ ./main_
```

Conclusión:

Los editores de texto son la principal herramienta que utiliza un programador para escribir código, editarlo y compartirlo.