



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor :* Alejandro Pimentel

*Asignatura:* Fundamentos de Programación

*Grupo:* 3

*No de Práctica(s):* 6

*Integrante(s):* Meraz Dionicio Israel

*No. de Equipo de  
cómputo empleado:* 2

*No. de Lista o Brigada:* 8875

*Semestre:* 1

*Fecha de entrega:* 30 de Septiembre de 2019

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## 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.

## INTRODUCCION

El texto plano es un tipo de archivo más sencillo que hay, ya que en su contenido no hay otra cosa más que el texto que lo conforma.

Un editor de texto es un programa capaz de editar texto plano, no se pueden confundir con un procesador de texto.

- Hacer una investigación acerca de los siguientes tipos de archivos.

**TXT**, Constituye el formato básico, pues solamente presenta texto, sin ningún tipo de “enriquecimiento” visual más como tipos o tamaños de letra, ni permite mezclar imágenes con el texto, ni siquiera funcionalidades tan simples para los demás procesadores y formatos de texto como el justificar a la derecha. Sencillamente es texto, eso sí, altamente portable -aunque no exento de problemas en este sentido- entre sistemas operativos distintos.

<https://www.definicionabc.com/tecnologia/formatos-texto.php>

**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 estuviésemos empleando directamente HTML. En otras palabras es una herramienta de software que convierte el lenguaje en HTML válido.

<https://www.genbeta.com/guia-de-inicio/que-es-markdown-para-que-sirve-y-como-usarlo>

**HTML**; es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de las siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, no es un lenguaje de programación. Y se usa para crear y representar visualmente una página web.

<https://developer.mozilla.org/es/docs/Web/HTML>

**LaTeX**, Es un sistema de composición de textos que está orientado especialmente a la creación de documentos científicos que contengan formulas matemáticas, cuadros y tablas. Además, también se pueden crear otros tipos de documentos, que pueden ser desde cartas sencillas hasta libros completos. LATEX está organizado sobre TEX.

<https://www.ecured.cu/LaTeX>

**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.

[https://techlandia.com/archivo-extension-csv-hechos\\_47208/](https://techlandia.com/archivo-extension-csv-hechos_47208/)

- Seguir el tutor de vim o vi.

```

MINGW64/c/Users/ACER/Downloads
ACER@EQUIPO MINGW64 ~ (master)
$ cd Downloads/
ACER@EQUIPO MINGW64 ~/Downloads (master)
$ vi vim-tutor.txt

```

Primero ubica la carpeta en donde descargaste el tutor y después ingresa con vi y el nombre del archivo.

```

ii NOTA: Antes de ejecutar alguno de los pasos siguientes lea primero
la lección entera!!

1. Pulse la tecla <ESC> (para asegurarse de que está en modo Normal).

2. Escriba:                :q! <INTRO>

---> Esto provoca la salida del editor SIN guardar ningún cambio que se haya
hecho. Si quiere guardar los cambios y salir escriba:
                :wq <INTRO>

3. Cuando vea el símbolo del sistema, escriba el mandato que le trajo a este
tutor. Este puede haber sido:    vimtutor <INTRO>
Normalmente se usaría:         vim tutor <INTRO>

---> 'vim' significa entrar al editor, 'tutor' es el fichero a editar.

4. Si ha memorizado estos pasos y se se siente con confianza, ejecute los
pasos 1 a 3 para salir y volver a entrar al editor. Después mueva el
cursor hasta la Lección 1.3.

vim-tutor.txt [unix] (12:54 29/09/2019)          54,0-1  6%
:wq

```

Para salir del tutor de vi.  
Es escribir: q! para guardar sin cambios  
:wq para guardar con cambios

```

MINGW64/c/Users/ACER/Downloads
Lección 1.3: EDICIÓN DE TEXTO - BORRADO

** Estando en modo Normal pulse  x  para borrar el carácter sobre el cursor. **

1. Mueva el cursor a la línea de abajo señalada con --->.
2. Para corregir los errores, mueva el cursor hasta que esté bajo el
carácter que va a ser borrado.
3. Pulse la tecla  x  para borrar el carácter sobrante.
4. Repita los pasos 2 a 4 hasta que la frase sea la correcta.
---> La vaca saltó sobre la luna.
5. Ahora que la línea esta correcta, continúe con la Lección 1.4.

NOTA: A medida que vaya avanzando en este tutor no intente memorizar,
aprenda practicando.
vim-tutor.txt[+] [unix] (12:54 29/09/2019)          86,0-1  9%

```

```

Lección 1.4: EDICIÓN DE TEXTO - INSERCIÓN

** Estando en modo Normal pulse  i  para insertar texto. **

1. Mueva el cursor a la primera línea de abajo señalada con --->.
2. Para que la primera línea se igual a la segunda mueva el cursor bajo el
primer carácter que sigue al texto que ha de ser insertado.
3. Pulse  i  y escriba los caracteres a añadir.
4. A medida que sea corregido cada error pulse <ESC> para volver al modo
Normal. Repita los pasos 2 a 4 para corregir la frase.
---> Falta algo de texto en esta línea.
---> Falta algo de texto en esta línea.
5. Cuando se sienta cómodo insertando texto pase al resumen que esta más
abajo.

vim-tutor.txt[+] [unix] (12:54 29/09/2019)          109,39-38 12%

```

```

MINGW64/c/Users/ACER/Downloads
** 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.

Lección 2.2: MAS MANDATOS PARA BORRAR

vim-tutor.txt[+] [unix] (12:54 29/09/2019)          159,45 19%

```

```

Lección 2.2: MAS 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 (DESPUES del primer . ).
4. Escriba  d$  para borrar hasta el final de la línea.
---> Alguien ha escrito el final de esta línea dos veces.

vim-tutor.txt[+] [unix] (12:54 29/09/2019)          177,59-58 21%

```

**\*\* 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 azucar es dulce,
- 7) Y así eres tu.

#### Lección 2.5: EL MANDATO DESHACER

vim-tutor.txt[+] [unix] (12:54 29/09/2019)

224,7 27%

**\*\* 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.

#### RESUMEN DE LA LECCIÓN 2

vim-tutor.txt[+] [unix] (12:54 29/09/2019)

245,62-61 30%

**\*\* 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.

#### RESUMEN DE LA LECCIÓN 2

vim-tutor.txt[+] [unix] (12:54 29/09/2019)

245,9 30%

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,
  - b) La inteligencia se aprende,
  - d) ¿Puedes aprenderla tu?

#### 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.

vim-tutor.txt[+] [unix] (12:54 29/09/2019)

293,6 38%

#### 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.

---> ¡Cuando esta línea fue escrita alguien pulsó algunas teclas equivocadas!  
---> ¡Cuando esta línea fue escrita alguien pulsó algunas teclas equivocadas!

vim-tutor.txt[+] [unix] (12:54 29/09/2019)

311,73-70 39%

**\*\* 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 'ínea').
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.

#### Lección 3.4: MAS CAMBIOS USANDO c

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

vim-tutor.txt[+] [unix] (12:54 29/09/2019)

337,63-62 43%

-- INSERTAR --

MINGW64/C:/Users/ACER/Downloads

**\*\* 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 'ínea').
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.

#### Lección 3.4: MAS CAMBIOS USANDO c

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

vim-tutor.txt[+] [unix] (12:54 29/09/2019)

337,62-61 43%

1. El mandato change funciona de la misma forma que delete. El formato es:  
[número] c objeto O 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\$.

#### RESUMEN DE LA LECCIÓN 3

vim-tutor.txt[+] [unix] (12:54 29/09/2019)

360,74-73 46%

Para lecturas y estudios posteriores se recomienda el libro:  
Learning the Vi Editor - por Linda Lamb  
Editorial: O'Reilly & Associates Inc.  
Es un buen libro para llegar a saber casi todo lo que desee hacer con Vi.  
La sexta edición incluye también información sobre Vim.

Este tutorial ha sido escrito por Michael C. Pierce y Robert K. Ware,  
Colorado School of Mines utilizando ideas suministradas por Charles Smith,  
Colorado State University.  
E-mail: bware@mines.colorado.edu.

Modificado para Vim por Bram Moolenaar.

Traducido del inglés por:

Eduardo F. Amatria  
Correo electrónico: efernal@platea.pntic.mec.es

Pase ahora a la lección siguiente.

#### Lección 4.1: SITUACIÓN EN EL FICHERO Y SU ESTADO

**\*\* Pulse CTRL-g para mostrar su situación en el fichero y su estado.  
Pulse MAYU-G para moverse a una determinada línea del fichero. \*\***

Nota: ¡¡Lea esta lección entera antes de ejecutar alguno de los pasos!!

1. Mantenga pulsada la tecla Ctrl y pulse g. Aparece una línea de estado al final de la pantalla con el nombre del fichero y la línea en la que está situado. Recuerde el número de la línea para el Paso 3.
2. Pulse Mayu-G para ir al final del fichero.
3. Escriba el número de la línea en la que estaba y después Mayu-G. Esto le volverá a la línea en la que estaba cuando pulsó Ctrl-g.  
(Cuando escriba los números NO se mostrarán en la pantalla).

vim-tutor.txt [unix] (12:54 29/09/2019) 388,0-1 51%  
"vim-tutor.txt" 769 lines --50%--

## La situación del fichero y su estado

#### Lección 4.2: EL MANDATO «SEARCH» (buscar)

**\*\* Escriba / seguido de una frase para buscar la frase. \*\***

1. En modo Normal pulse el carácter /. Fijese que tanto el carácter / como el cursor aparecen en la última línea de la pantalla, lo mismo que el mandato :.
2. Escriba ahora errroor <INTRO>. Esta es la palabra que quiere buscar.
3. Para repetir la búsqueda, simplemente pulse n.  
Para buscar la misma frase en la dirección opuesta, pulse Mayu-N.
4. Si quiere buscar una frase en la dirección opuesta (hacia arriba), utilice el mandato ? en lugar de /.

---> Cuando la búsqueda alcanza el final del fichero continuará desde el principio.

«errroor» no es la forma de deletrear error; errroor es un error.

vim-tutor.txt [unix] (12:54 29/09/2019) 417,22 54%  
/errroor

#### Lección 4.2: EL MANDATO «SEARCH» (buscar)

**\*\* Escriba / seguido de una frase para buscar la frase. \*\***

1. En modo Normal pulse el carácter /. Fijese que tanto el carácter / como el cursor aparecen en la última línea de la pantalla, lo mismo que el mandato :.
2. Escriba ahora errroor <INTRO>. Esta es la palabra que quiere buscar.
3. Para repetir la búsqueda, simplemente pulse n.  
Para buscar la misma frase en la dirección opuesta, pulse Mayu-N.
4. Si quiere buscar una frase en la dirección opuesta (hacia arriba), utilice el mandato ? en lugar de /.

---> Cuando la búsqueda alcanza el final del fichero continuará desde el principio.

«errroor» no es la forma de deletrear error; errroor es un error.

vim-tutor.txt [unix] (12:54 29/09/2019) 417,22 54%  
?errroor

Funciona para buscar palabras en el texto.

#### Lección 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS

**\*\* 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 debería 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.

vim-tutor.txt [unix] (12:54 29/09/2019) 447,11 57%

#### Lección 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS

**\*\* 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 debería 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.

vim-tutor.txt [unix] (12:54 29/09/2019) 447,44-43 57%

#### Lección 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS

**\*\* 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 debería 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.



```
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.

---> Laas mejores épocas para ver laas flores son laas primaveras.

4. Para cambiar todas las apariciones de una expresión ente 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.

vim-tutor.txt [unix] (12:54 29/09/2019) 464,36-35 60%
:s/laas/las
```

```
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.

4. Para cambiar todas las apariciones de una expresión ente 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.

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 464,1 60%
:s/laas/las
```

```
var
vcredist_arm.exe
Vega,+Lope+De+--+Fuenteovejuna.pdf
VID_20180413_183315.3gp
VID_20180418_184825.3gp
VID_20180420_184653.3gp
VID_20180427_185537.3gp
'Vida-retirada (1).docx'
Vida-retirada.docx
vim-tutor.txt
VIRREINATO
'virtual-dj-7.0 (1).exe'
virtual-dj-7.0.exe
virtual-dj-8-5-3573.msi
'WhatsApp Image 2018-10-17 at 9.41.18 PM.jpeg'
'WhatsApp Image 2019-01-13 at 7.06.50 PM (1).jpeg'
'while (1).docx'
while.docx
winupcompat.diagcab
'Wiz Khalifa - See You Again ft. Charlie Puth [Official Video] Furious 7 Soundtr
ack.mp3'
yo-soy-132.docx

Pulse INTRO o escriba una orden para continuar
```

```
Lección 5.2: MAS 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 sería una copia exacta del tutor cuando lo
ha guardado.

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 536,1 69%
"TEST" [Nuevo][UNIX] 769L, 28038C escritos
```

Como ejecutar un mandato externo.

```
Lección 5.3: UN MANDATO DE ESCRITURA SELECTIVO

** Para guardar parte del fichero escriba :#,# NOMBRE_DEL_FICHERO **

1. Escriba de nuevo, una vez más, :!dir o :!ls para obtener una lista
de su directorio y elija nombre de fichero adecuado, como TEST.

2. Mueva el cursor al principio de la pantalla y pulse Ctrl-g para saber
el número de la línea correspondiente. ¡RECUERDE ESTE NÚMERO!

3. Ahora mueva el cursor a la última línea de la pantalla y pulse Ctrl-g
de nuevo. ¡RECUERDE TAMBIEN ESTE NÚMERO!

4. Para guardar SOLAMENTE una parte de un fichero, escriba :#,# w TEST
donde #,# son los números que usted ha recordado (primera línea,
última línea) y TEST es su nombre de fichero.

5. De nuevo, vea que el fichero esta ahí con :!dir pero NO lo borre.

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 558,1 72%
"vim-tutor.txt" [Modificado] 769 líneas --72%--
```

```
3. Ahora recupere el fichero TEST utilizando el mandato :r TEST donde
TEST es el nombre del fichero.

NOTA: El fichero recuperado se sitúa a partir de la posición del cursor.

1. Escriba de nuevo, una vez más, :!dir o :!ls para obtener una lista
de su directorio y elija nombre de fichero adecuado, como TEST.

2. Mueva el cursor al principio de la pantalla y pulse Ctrl-g para saber
el número de la línea correspondiente. ¡RECUERDE ESTE NÚMERO!

3. Ahora mueva el cursor a la última línea de la pantalla y pulse Ctrl-g
de nuevo. ¡RECUERDE TAMBIÉN ESTE NÚMERO!

4. Para guardar SOLAMENTE una parte de un fichero, escriba :#,# w TEST
donde #,# son los números que usted ha recordado (primera línea,
última línea) y TEST es su nombre de fichero.

5. De nuevo, vea que el fichero esta ahí con :!dir pero NO lo borre.
4. Para verificar que el fichero ha sido recuperado, mueva el cursor hacia
arriba y vea que hay dos copias de la Lección 5.3, la original y la
versión del fichero.

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 582,0-1 75%
```

De alguna copie copie un fichero y lo peque en otro

```
Lección 6.1: EL MANDATO «OPEN» (abrir)

** Pulse o para abrir una línea debajo del cursor
y situarle en modo Insert **

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

2. Pulse o (minúscula) para abrir una línea por DEBAJO del cursor
y situarle en modo Insert.

3. Ahora copie la línea señalada con ---> y pulse <ESC> para salir del
modo Insert.

---> Luego de pulsar o el cursor se sitúa en la línea abierta en modo Insert.
---> Luego de pulsar o el cursor se situa en la línea abierta en modo Insert.
4. Para abrir una línea por encima del cursor, simplemente pulse una o
mayúscula, en lugar de una o minúscula. Pruebe este en la línea siguiente.
Abra una línea sobre ésta pulsando Mayu-O cuando el curso está en esta línea.

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 641,81-80 81%
-- INSERTAR --
```

```
1. Mueva el cursor al final de la primera línea de abajo señalada con --->
pulsando $ en modo Normal.

2. Escriba una a (minúscula) para añadir texto DESPUÉS del carácter
que está sobre el cursor. (A mayúscula añade texto al final de la línea).

Nota: ¡Esto evita el pulsar i , el último carácter, el texto a insertar,
<ESC>, cursor a la derecha y, finalmente, x , sólo para añadir algo
al final de una línea!

3. Complete ahora la primera línea. Nótese que append es exactamente lo
mismo que modo Insert, excepto por el lugar donde se inserta el texto.

---> Esta línea le permitirá practicar
---> Esta línea le permitirá practicar el añadido de texto al final de una línea.

Lección 6.3: OTRA VERSIÓN DE «REPLACE» (reemplazar)

** Pulse una R mayúscula para sustituir más de un carácter. **

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 665,40-38 85%
-- INSERTAR --
```

El o funciona como un tabulador

Antes

```
1. Mueva el cursor al final de la primera línea de abajo señalada con --->
pulsando $ en modo Normal.

2. Escriba una a (minúscula) para añadir texto DESPUÉS del carácter
que está sobre el cursor. (A mayúscula añade texto al final de la línea).

Nota: ¡Esto evita el pulsar i , el último carácter, el texto a insertar,
<ESC>, cursor a la derecha y, finalmente, x , sólo para añadir algo
al final de una línea!

3. Complete ahora la primera línea. Nótese que append es exactamente lo
mismo que modo Insert, excepto por el lugar donde se inserta el texto.

---> Esta línea le permitirá practicar el añadido de texto al final de una línea.
---> Esta línea le permitirá practicar el añadido de texto al final de una línea.

Lección 6.3: OTRA VERSIÓN DE «REPLACE» (reemplazar)

** Pulse una R mayúscula para sustituir más de un carácter. **

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 665,85-81 85%
-- INSERTAR --
```

Después

```
Lección 6.3: OTRA VERSIÓN DE «REPLACE» (reemplazar)

** Pulse una R mayúscula para sustituir más de un carácter. **

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

2. Sitúe el cursor al comienzo de la primera palabra que sea diferente
de las de la segunda línea marcada con ---> (la palabra 'anterior').

3. Ahora pulse R y sustituya el resto del texto de la primera línea
escribiendo sobre el viejo texto para que la primera línea sea igual
que la primera.

---> Para hacer que esta línea sea igual que la siguiente escribiateclas.
---> Para hacer que esta línea sea igual que la siguiente escriba R y el texto.

4. Nótese que cuando pulse <ESC> para salir, el texto no alterado permanece.

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 683,67-66 87%
-- REEMPLAZAR --
```

```
Lección 6.3: OTRA VERSIÓN DE «REPLACE» (reemplazar)

** Pulse una R mayúscula para sustituir más de un carácter. **

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

2. Sitúe el cursor al comienzo de la primera palabra que sea diferente
de las de la segunda línea marcada con ---> (la palabra 'anterior').

3. Ahora pulse R y sustituya el resto del texto de la primera línea
escribiendo sobre el viejo texto para que la primera línea sea igual
que la primera.

---> Para hacer que esta línea sea igual que la siguiente escriba R y el texto.
---> Para hacer que esta línea sea igual que la siguiente escriba R y el texto.

4. Nótese que cuando pulse <ESC> para salir, el texto no alterado permanece.

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 683,81-80 87%
-- REEMPLAZAR --
```

```
** Fijar una opción de forma que una búsqueda o sustitución ignore la caja **
(Para el concepto de caja de una letra, véase la nota al final del fichero)

1. Busque ignorar' introduciendo:
/ignorar
Repita varias veces la búsqueda pulsando la tecla n

2. Fije la opción 'ic' (Ignorar la caja de la letra) escribiendo:
:set ic

3. Ahora busque 'ignorar' de nuevo pulsando n
Repita la búsqueda varias veces más pulsando la tecla n

4. Fije las opciones 'hlsearch' y 'insearch':
:set hls is

5. Ahora introduzca la orden de búsqueda otra vez, y vea qué pasa:
/ignore

RESUMEN DE LA LECCIÓN 6

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 697,12 90%
:set ic
```

```
** Fijar una opción de forma que una búsqueda o sustitución ignore la caja **
(Para el concepto de caja de una letra, véase la nota al final del fichero)

1. Busque ignorar' introduciendo:
/ignorar
Repita varias veces la búsqueda pulsando la tecla n

2. Fije la opción 'ic' (Ignorar la caja de la letra) escribiendo:
:set ic

3. Ahora busque 'ignorar' de nuevo pulsando n
Repita la búsqueda varias veces más pulsando la tecla n

4. Fije las opciones 'hlsearch' y 'insearch':
:set hls is

5. Ahora introduzca la orden de búsqueda otra vez, y vea qué pasa:
/ignore

RESUMEN DE LA LECCIÓN 6

vim-tutor.txt[+] [unix] (12:54 29/09/2019) 697,13 90%
:set hls is
```

```
(Para el concepto de caja de una letra, véase la nota al final del fichero)

1. Busque ignorar' introduciendo:
   /ignorar
   Repita varias veces la búsqueda pulsando la tecla n

2. Fije la opción 'ic' (Ignorar la caja de la letra) escribiendo:
   :set ic

3. Ahora busque 'ignorar' de nuevo pulsando n
   Repita la búsqueda varias veces más pulsando la tecla n

4. Fije las opciones 'hlsearch' y 'insearch':
   :set hls is

5. Ahora introduzca la orden de búsqueda otra vez, y vea qué pasa:
   /ignore

-----
RESUMEN DE LA LECCIÓN 6
vim-tutor.txt[+] [unix] (12:54 29/09/2019) 711,7 90%
/ignore
```

La última lección que es la 7, en general se trata de saber la manera de cómo se escribe un comando de ayuda.

```
help.txt For Vim version 8.1. Last change: 2019 Jul 21

VIM - main help file

Move around: Use the cursor keys, or "h" to go left,          h k
              "j" to go down, "k" to go up, "l" to go right.    j
Close this window: Use ":q<Enter>".
Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!).

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-].
With the mouse: Use ":set mouse=a" to enable the mouse (in xterm or GUI).
                  Double-click the left mouse button on a tag, e.g. bars.
Jump back: Type CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the :help command.
                  Prepend something to specify the context: help-context

WHAT      PREPEND  EXAMPLE
Normal mode command      :help x
help.txt[Ayuda][-][RO] [unix] (08:41 17/08/2019) 1,1 Comienzo

:help w <INTRO>
vim-tutor.txt[+] [unix] (12:54 29/09/2019) 753,7 95%
"help.txt" [Solo lectura][UNIX] 238L, 8894C
```

```
help.txt For Vim version 8.1. Last change: 2019 Jul 21

VIM - main help file

Move around: Use the cursor keys, or "h" to go left,          h k
              "j" to go down, "k" to go up, "l" to go right.    j
Close this window: Use ":q<Enter>".
Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!).

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-].
With the mouse: Use ":set mouse=a" to enable the mouse (in xterm or GUI).
                  Double-click the left mouse button on a tag, e.g. bars.
Jump back: Type CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the :help command.
                  Prepend something to specify the context: help-context

WHAT      PREPEND  EXAMPLE
Normal mode command      :help x
help.txt[Ayuda][-][RO] [unix] (08:41 17/08/2019) 1,1 Comienzo

:help w <INTRO>
vim-tutor.txt[+] [unix] (12:54 29/09/2019) 753,7 95%
:q
```

Al pulsar la tecla F1

Escribir :help

Si no quieres todo el manual de ayuda y solo buscar ciertas especificaciones puedes escribir :help mas el mandato, por ejemplo;

```
vim-tutor.txt[+] [unix] (12:54 29/09/2019) 743,7 94%
motion.txt[Ayuda][-][RO] [unix] (07:10 24/06/2019) 370,30-69 27%
de estas tres formas:
- pulse la tecla <AYUDA> (si dispone de ella)
vim-tutor.txt[+] [unix] (12:54 29/09/2019) 743,7 94%
motion.txt [Solo lectura][UNIX] 1327L, 50875C

4. word motions word-motions
<S-Right> or <S-Right> |v [count] words forward. exclusive motion.
<C-Right> or <C-Right> W [count] WORDS forward. exclusive motion.
Forward to the end of word [count] inclusive. Does not stop in an empty line.
Forward to the end of WORD [count] inclusive. Does not stop in an empty line.
<S-Left> or <S-Left> b [count] words backward. exclusive motion.
<C-Left> or <C-Left> B [count] WORDS backward. exclusive motion.
Backward to the start of word [count] inclusive. Does not stop in an empty line.
Backward to the start of WORD [count] inclusive. Does not stop in an empty line.
```

W

```
When showing file names, directories are highlighted (see 'highlight' option). Names where 'suffixes' matches are moved to the end.
The 'wildoptions' option can be set to "tagfile" to list the file of matching tags.
CTRL-I c_wildchar f_<Tab>

'wildchar' option
A match is done on the pattern in front of the cursor. The match (if there are several, the first match) is inserted in place of the pattern. (Note: does not work inside a macro, because <Tab> or <Esc> are mostly used as 'wildchar', and these have a special meaning in some macros.) When typed again and there were multiple matches, the next match is inserted. After the last match, the first is used again (wrap around).
The behavior can be changed with the 'wildmode' option.
C_<S-Tab>
Like 'wildchar' or <Tab>, but begin with the last match and then go to the previous match.
<S-Tab> does not work everywhere.
cmdline.txt[Ayuda][-][RO] [unix] (07:10 24/06/2019) 397,30-65 33%
de estas tres formas:
- pulse la tecla <AYUDA> (si dispone de ella)
vim-tutor.txt[+] [unix] (12:54 29/09/2019) 743,7 94%
cmdline.txt [Solo lectura][UNIX] 1202L, 48412C
```

c\_<T

```
For an overview of built-in functions see functions.
For a list of Vim variables see vim-variable.
For a complete listing of all help items see help-tags.

1. Insert mode insert-index
tag char action in Insert mode
i_CTRL-@ CTRL-@ insert previously inserted text and stop insert
i_CTRL-A CTRL-A insert previously inserted text not used i_CTRL-B-gone
i_CTRL-C CTRL-C quit insert mode, without checking for abbreviation, unless 'insertmode' set.
i_CTRL-D CTRL-D delete one shiftwidth of indent in the current line
i_CTRL-E CTRL-E insert the character which is below the cursor not used (but by default it's in 'cinkeys' to re-indent the current line)
i_CTRL-F CTRL-F
index.txt[Ayuda][-][RO] [unix] (14:25 03/07/2019) 32,21-57 1%
de estas tres formas:
- pulse la tecla <AYUDA> (si dispone de ella)
vim-tutor.txt[+] [unix] (12:54 29/09/2019) 743,7 94%
"index.txt" [Solo lectura][UNIX] 1728L, 79491C
```

Insert-index



## Notepad ++

A screenshot of the Notepad++ text editor. The title bar shows 'main.c' and 'UNREGISTERED'. The editor window displays a C program with the following code:

```
1 int main()
2 {
3     printf("Mi primer programa\n");
4
5     return 0;
6 }
```

Se utilizó el programa sublime.text que es parecido al notepad++, lo primero que hay que hacer es guardarlo en .c para que el editor de texto plano lo reconozca como en lenguaje en c

La \n significa un salto de línea, esto para que no lo imprima muy pegado a la entrada de la terminal , y te quede como el siguiente ejemplo

A screenshot of a terminal window showing the compilation and execution of the C program. The output is as follows:

```
main.c
[Kuwait02:Downloads fp03alu29$ gcc main.c -o main_
main.c:3:2: warning: implicitly declaring library function 'printf' with type
'int (const char *, ...)' [-Wimplicit-function-declaration]
    printf("Mi primer programa\n");
    ^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
declaration for 'printf'
main.c:6:2: warning: treating Unicode character as whitespace
[-Wunicode-whitespace]
}
main.c:6:4: warning: treating Unicode character as whitespace
[-Wunicode-whitespace]
}
3 warnings generated.
[Kuwait02:Downloads fp03alu29$ ./main_
-bash: ./main: No such file or directory
[Kuwait02:Downloads fp03alu29$ ls
main.c main_
[Kuwait02:Downloads fp03alu29$ ./main_
Mi primer programa
[Kuwait02:Downloads fp03alu29$ ]
```

Los warnings no son alarmantes ya que no nombraste a una función pero si la puede leer.

A screenshot of a terminal window showing the compilation and execution of the C program with an additional print statement. The output is as follows:

```
Kuwait02:Downloads fp03alu29$ ./main_
Mi primer programa
[Kuwait02:Downloads fp03alu29$ ls
Práctica 6 main.c main_
[Kuwait02:Downloads fp03alu29$ gcc main.c -o main_
main.c:3:2: warning: implicitly declaring library function 'printf' with type
'int (const char *, ...)' [-Wimplicit-function-declaration]
    printf("Mi primer programa\n" "Por primera vez\n");
    ^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
declaration for 'printf'
main.c:6:2: warning: treating Unicode character as whitespace
[-Wunicode-whitespace]
}
main.c:6:4: warning: treating Unicode character as whitespace
[-Wunicode-whitespace]
}
3 warnings generated.
[Kuwait02:Downloads fp03alu29$ ./main_
Mi primer programa
Por primera vez
[Kuwait02:Downloads fp03alu29$ ]
```

## CONCLUSION:

Hemos observado lo que puede significar un texto plano y que existen varios tipos de archivos en texto plano como el txt o el html y sus diferencias, hemos conocido sobre un editor de texto que es el vi o vim que está en la terminal de Linux, es complicado al principio por no estar tan relacionado con nosotros y apenas lo estamos conociendo pero serian una buena oportunidad para no tenerle miedo a la computadora, ya por terminar entramos a la programación en c con un simple programa.