



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: MC. Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): Práctica 6

Integrante(s): Martínez Marcelino Dalila

*No. de Equipo de
cómputo empleado:* No. de cuenta: 313080119

No. de Lista o Brigada:

Semestre: 2020-1

Fecha de entrega: Lunes 30 de septiembre de 2019

Observaciones:

CALIFICACIÓN: _____

PRÁCTICA NO. 6

Introducción.

En esta práctica se conocerá los diferentes tipos de archivo en donde se maneja el texto plano. También comenzaremos con el uso de herramientas como son los editores de texto para el desarrollo de programas en Lenguaje C. Por último, el estudiante hará un programa de prueba con alguno de los editores y lo compilará.






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.

Actividad 1.

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
-  MARKDOWN
-  HTML
-  LaTeX
-  CSV

-  TXT

Documentos de texto plano guardados en formato TXT se pueden crear, abrir y editar utilizando una amplia variedad de programas de procesamiento de texto y de edición de textos desarrollados para sistemas Linux, ordenadores y plataformas Mac Microsoft basado en Windows. El contenido de estos [.txt](#) archivos de texto ASCII (acrónimo inglés de American Standard Code for Information Interchange —Código Estándar Estadounidense para el Intercambio de Información—), sin formato es que se pueden guardar como [.txt](#) documentos en archivos de tamaño reducido.

-  MARKDOWN

Markdown fue desarrollado en 2004 por [John Gruber](#), y se refiere tanto a (1) una manera de formar archivos de texto, como a (2) una utilidad del lenguaje de programación Perl para convertir archivos Markdown en HTML. Al utilizar la sintaxis de Markdown, se hace posible producir archivos que pueden ser legibles como texto plano y que a la vez estén listos para ser formados en otras plataformas. Muchos generadores de bitácoras y de sitios estáticos, así como sitios como

GitHub, también aceptan Markdown y traducen estos archivos a HTML para su visualización en la web. Además, herramientas como Pandoc pueden convertir archivos en o desde Markdown. Los archivos en Markdown se guardan con la extensión `.md` y se pueden abrir en un editor de texto como TextEdit, Notepad, Sublime Text o Vim. Muchos sitios web o plataformas de publicación también ofrecen editores basados en la web y/o extensiones para introducir texto utilizando la sintaxis de Markdown.

HTML

HTML significa Hypertext Markup Language, y estos archivos HTML se implementan en su mayoría en forma de páginas estáticas de sitios web. HTML se puede utilizar para integrar ciertos atributos de formato y las especificaciones de diseño en el contenido de estas páginas web. Hay una gran cantidad de aplicaciones que se pueden utilizar para crear y editar estos archivos HTML, y muchos de los navegadores web se pueden usar para abrir y ver el contenido de archivos en el `.html` formato. Los códigos HTML implementados en estos `.html` archivos son analizados por los navegadores web, lo que significa que está oculto a los espectadores. Microsoft Bloc de notas se puede utilizar para crear archivos HTML y también abrir y ver su código fuente.

LaTeX

Los archivos con la extensión TEX son documentos de texto creados por un programa llamado LaTeX. LaTeX es un software de composición tipográfica que permite al usuario crear documentos con distintos diagramas, tablas y esquemas científicos de alta calidad. A diferencia de los procesadores de texto habituales, como Microsoft Word o Apple Pages, los documentos TEX se utilizan principalmente en distintas ramas científicas. Se utilizan tanto para crear documentos y artículos sencillos como para preparar libros y otros documentos técnicos para su publicación. Los documentos TEX de LaTeX se utilizan en informática, física, distintas ingenierías, lingüística experimental y otros diversos campos debido a la alta calidad de los documentos resultantes y al soporte de distintos tipos de contenido que ofrecen. Los documentos TEX soportan la inclusión de expresiones matemáticas complejas, gráficos científicos y otros símbolos propios de materias concretas. LaTeX se desarrolló en 1985 y actualmente es mantenido, actualizado y desarrollado por el proyecto LaTeX3.

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.

Un fichero CSV es un archivo de texto que contiene una serie de valores separados por comas. Los valores pueden ser cualquier cosa, desde números de un presupuesto de una hoja de cálculo, hasta nombres y descripciones de una lista de clientes de un negocio.

Varios programas pueden crear y leer archivos CSV. Microsoft Excel utiliza extensivamente estos archivos, y Outlook puede exportar e importar archivos CSV para el transporte. Otras aplicaciones comunes que usan archivos CSV son OpenOffice, Microsoft Works, WordPerfect Office y NeoOffice.

Editores

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


Existen una variedad de editores como son, Sublime Text, Vim, Nano, Notepad++, Bluefish, Gedit, etc. En general, los editores difieren en su modo de uso y en las características que ofrecen, entre las características que ofrecen están: resaltado de palabras clave, autocompletado, autosangrado, integración de terminal, entre otros.

Para las practicas que se harán a partir de aquí en adelante yo he escogido utilizar estos dos tipos de editores:

 Sublime Text

 Vim

Los cuales me resultaron fácil de utilizar para desarrollar los programas de las prácticas.

 Vi o Vim (Editor de texto)

Es un 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.

Para lo cual se llevará acabo la siguiente actividad.

Actividad 2.

Seguir el tutor de vim.

Empezamos entrando en la terminal, una vez seguros que el archivo de vim está en la carpeta para poderlo ejecutar.

```
~/progra
DELL@DESKTOP-NJV5QTU ~
$ ls
progra

DELL@DESKTOP-NJV5QTU ~
$ cd progra

DELL@DESKTOP-NJV5QTU ~/progra
$ ls
ejemplo1.exe  ejemplo2.exe  ejemplo3.exe  prueba.c
ejemplo2.c    ejemplo3.c    primerprog.c  prueba.c~
ejemplo2.c~   ejemplo3.c~   primerprog.c~ vim-tutor.txt

DELL@DESKTOP-NJV5QTU ~/progra
$ vi vim-tutor.txt|
```

Entramos y seguimos las instrucciones que vienen.

```
~/progra
=====
= Bienvenido al tutor de VIM - VersiÃ³n 1.4 =
=====

Vim es un editor muy potente que dispone de muchos mandatos, demasiados
para ser explicados en un tutor como Ã©ste. Este tutor estÃ¡ diseÃ±ado
para describir suficientes mandatos para que usted sea capaz de
aprender fÃ¡cilmente a usar Vim como un editor de propÃ³sito general.

El tiempo necesario para completar el tutor es aproximadamente de 25-30
minutos, dependiendo de cuanto tiempo se dedique a la experimentaciÃ³n.

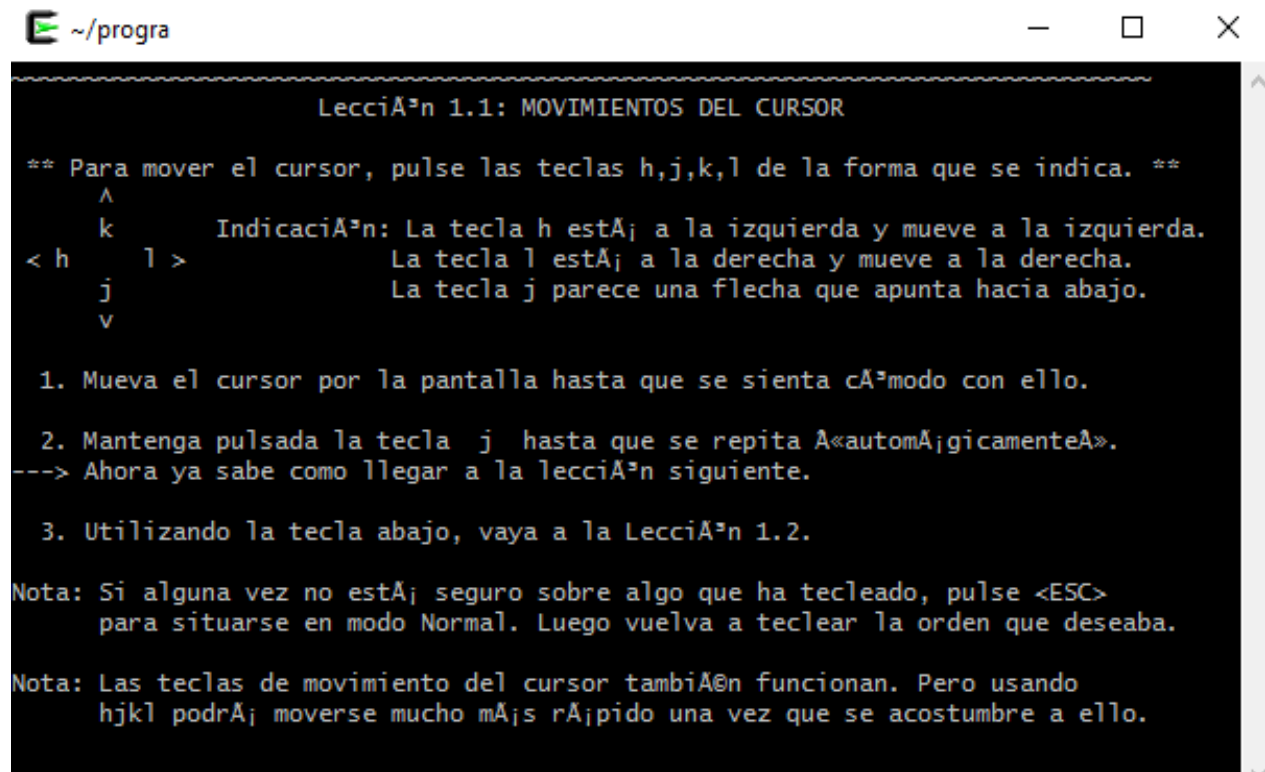
Los mandatos de estas lecciones modificarÃ¡n el texto. Haga una copia de
este fichero para practicar (con Ã©«vimtutorÃ© esto ya es una copia).

Es importante recordar que este tutor estÃ¡ pensado para enseÃ±ar con
la prÃ¡ctica. Esto significa que es necesario ejecutar los mandatos
para aprenderlos adecuadamente. Si sÃ³lo se lee el texto, se
olvidarÃ¡n los mandatos.

Ahora, asegÃºrese de que la tecla de bloqueo de mayÃºsculas no estÃ¡
activada y pulse la tecla j lo suficiente para mover el cursor
de forma que la LecciÃ³n 1.1 ocupe completamente la pantalla.

"vim-tutor.txt" 769 lines, 28041 characters
```

Practicamos un poco o mucho con los comandos de movimiento.



```
~/progra

Lección 1.1: MOVIMIENTOS DEL CURSOR

** Para mover el cursor, pulse las teclas h,j,k,l de la forma que se indica. **
      ^
      k      Indicaci3n: La tecla h est3 a la izquierda y mueve a la izquierda.
< h      l >      La tecla l est3 a la derecha y mueve a la derecha.
      j      La tecla j parece una flecha que apunta hacia abajo.
      v

1. Mueva el cursor por la pantalla hasta que se sienta c3modo con ello.

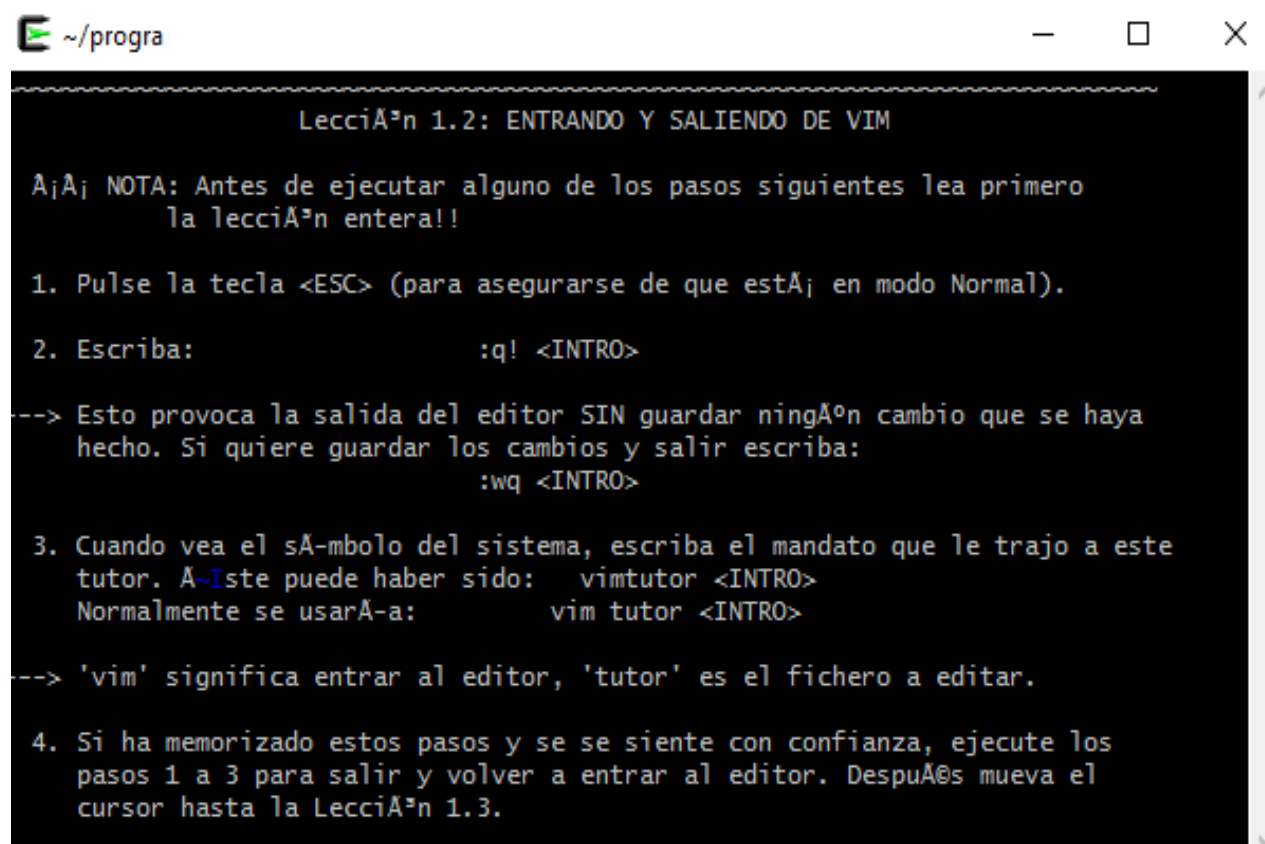
2. Mantenga pulsada la tecla j hasta que se repita A«autom3ticamenteA».
--> Ahora ya sabe como llegar a la lecci3n siguiente.

3. Utilizando la tecla abajo, vaya a la Lecci3n 1.2.

Nota: Si alguna vez no est3 seguro sobre algo que ha tecleado, pulse <ESC>
      para situarse en modo Normal. Luego vuelva a teclear la orden que deseaba.

Nota: Las teclas de movimiento del cursor tambi3n funcionan. Pero usando
      hjkl podr3 moverse mucho m3s r3pido una vez que se acostumbre a ello.
```

Practicamos a salir y entrar dentro del archivo.



```
~/progra

Lección 1.2: ENTRANDO Y SALIENDO DE VIM

A3A3 NOTA: Antes de ejecutar alguno de los pasos siguientes lea primero
      la lecci3n entera!!

1. Pulse la tecla <ESC> (para asegurarse de que est3 en modo Normal).

2. Escriba:                                :q! <INTRO>

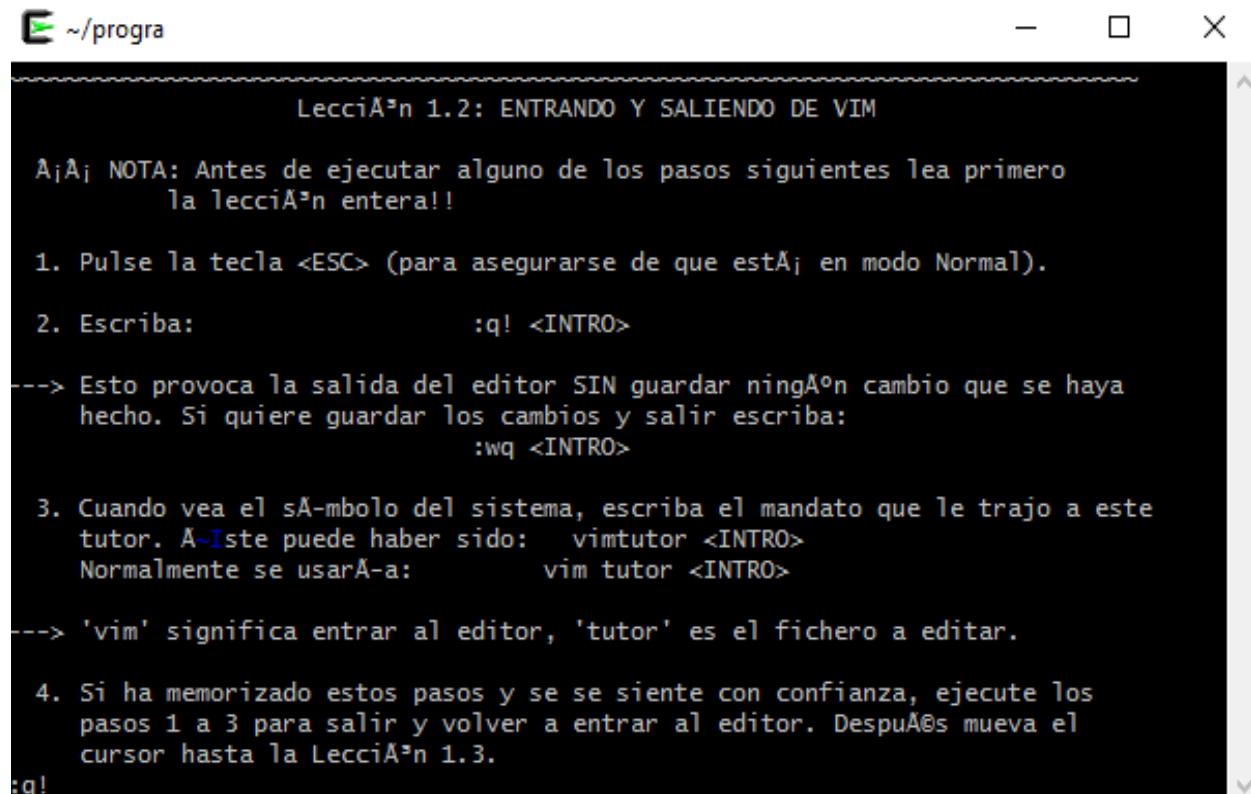
--> Esto provoca la salida del editor SIN guardar ning3n cambio que se haya
      hecho. Si quiere guardar los cambios y salir escriba:
                                :wq <INTRO>

3. Cuando vea el s3mbolo del sistema, escriba el mandato que le trajo a este
      tutor. A3ste puede haber sido: vimtutor <INTRO>
      Normalmente se usar3-a: 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. Despu3s mueva el
      cursor hasta la Lecci3n 1.3.
```

Vamos a salir



```
~/progra

Lección 1.2: ENTRANDO Y SALIENDO DE VIM

A¡A¡ 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>

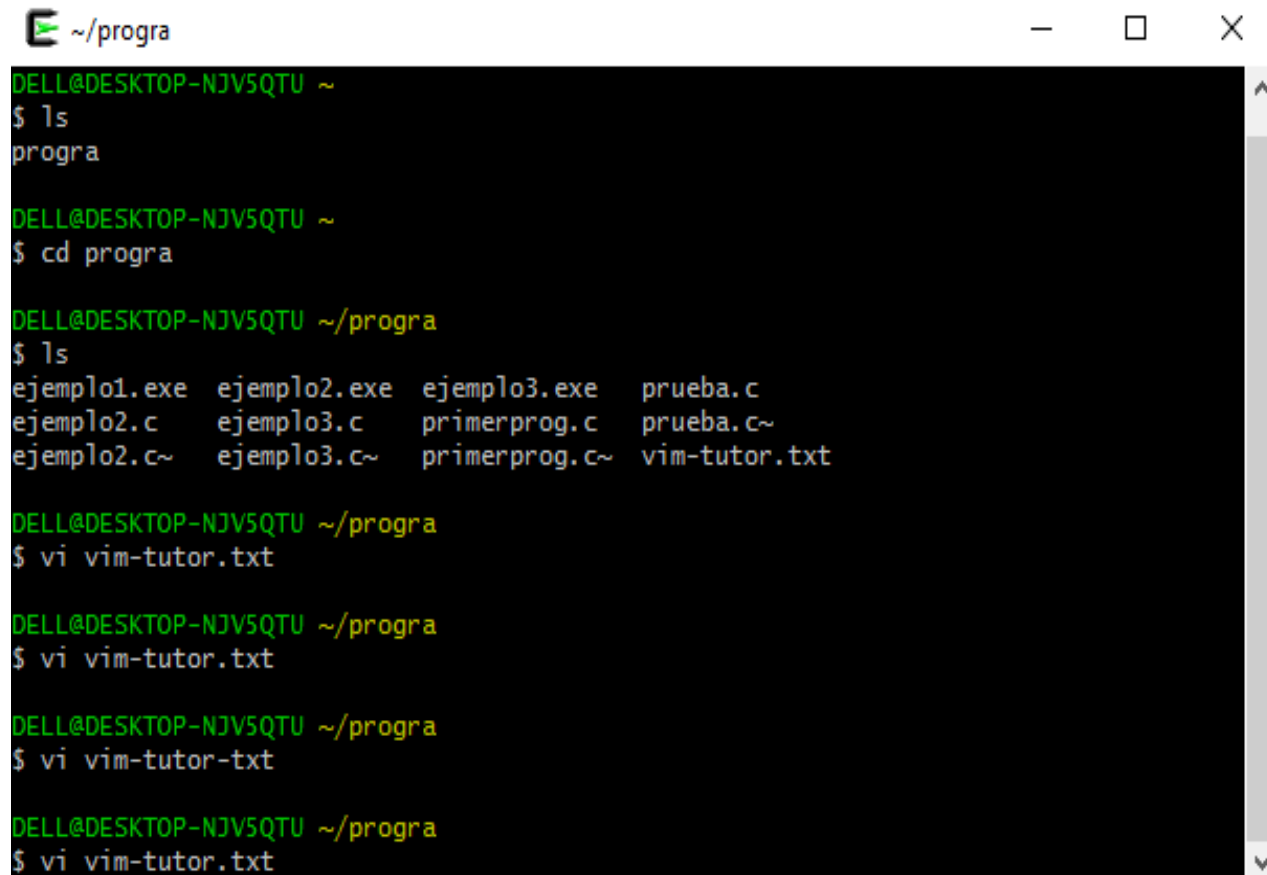
3. Cuando vea el símbolo del sistema, escriba el mandato que le trajo a este
tutor. A~Iste puede haber sido:  vimtutor <INTRO>
Normalmente se usar~a:          vim tutor <INTRO>

---> '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.

:q!
```

Vamos a entrar



```
~/progra

DELL@DESKTOP-NJV5QTU ~
$ ls
progra

DELL@DESKTOP-NJV5QTU ~
$ cd progra

DELL@DESKTOP-NJV5QTU ~/progra
$ ls
ejemplo1.exe  ejemplo2.exe  ejemplo3.exe  prueba.c
ejemplo2.c    ejemplo3.c    primerprog.c  prueba.c~
ejemplo2.c~   ejemplo3.c~   primerprog.c~ vim-tutor.txt

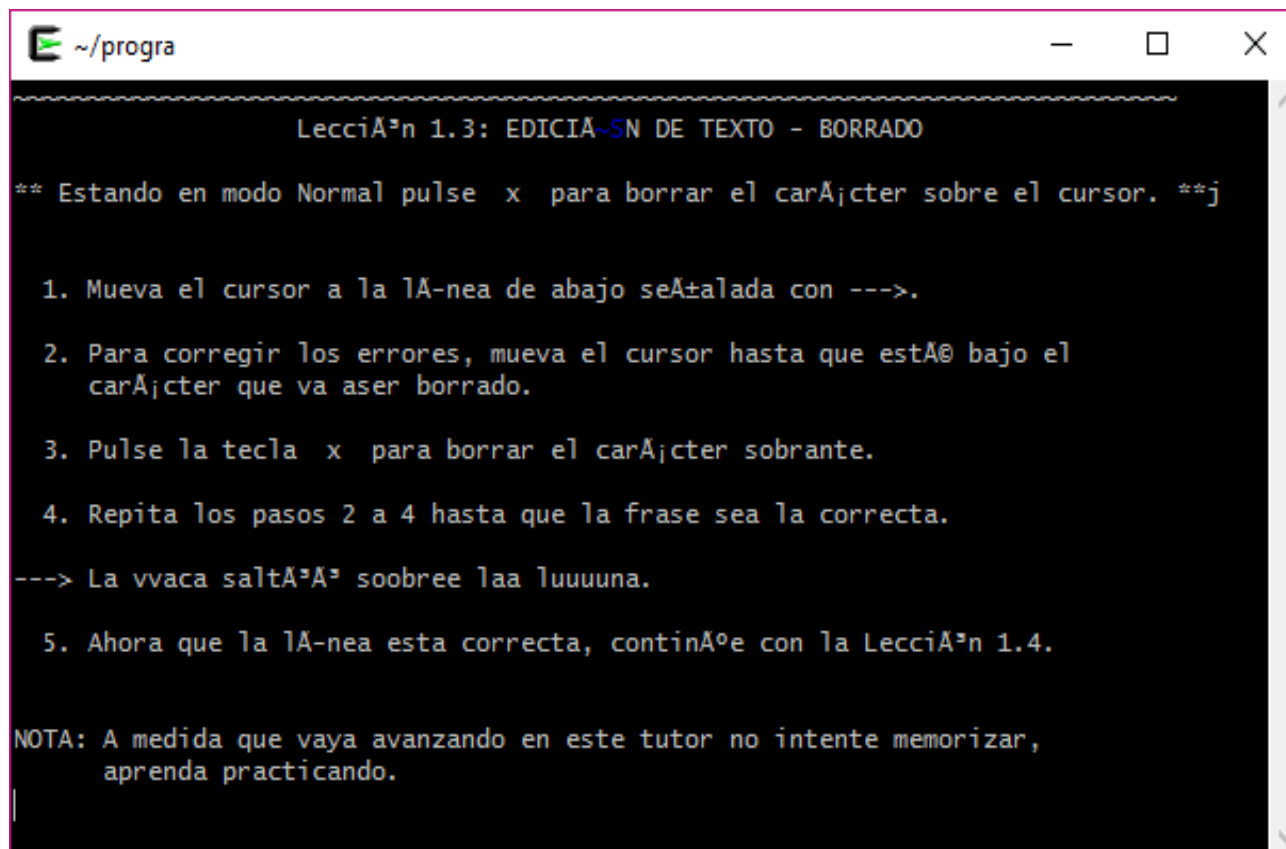
DELL@DESKTOP-NJV5QTU ~/progra
$ vi vim-tutor.txt

DELL@DESKTOP-NJV5QTU ~/progra
$ vi vim-tutor.txt

DELL@DESKTOP-NJV5QTU ~/progra
$ vi vim-tutor-txt

DELL@DESKTOP-NJV5QTU ~/progra
$ vi vim-tutor.txt
```

Aprendiendo a borrar en el archivo



The screenshot shows a terminal window with a title bar that includes a green icon and the text "~/progra". The terminal content is as follows:

```
~~~~~
Lección 1.3: EDICIÓN DE TEXTO - BORRADO

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

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.

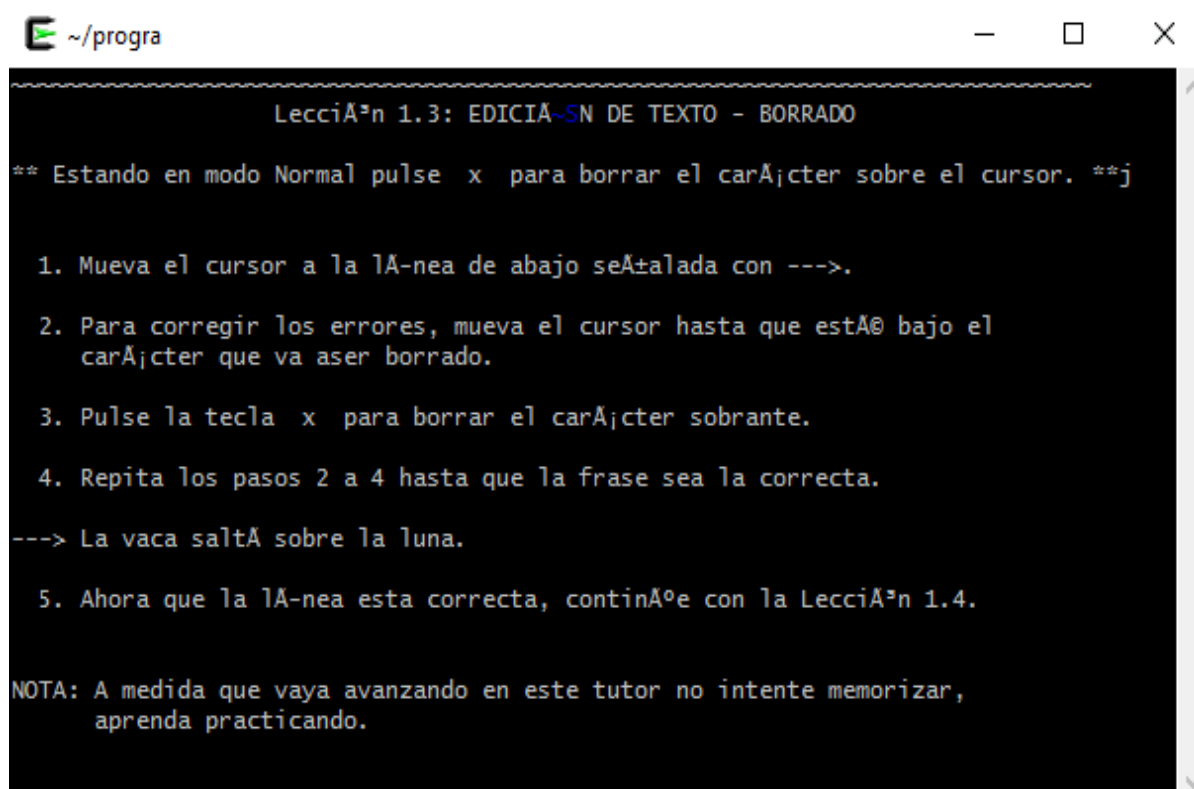
4. Repita los pasos 2 a 4 hasta que la frase sea la correcta.

---> 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.
|
```

Hacemos el ejercicio de la practica



The screenshot shows the same terminal window as above, but the text has been corrected. The terminal content is as follows:

```
~~~~~
Lección 1.3: EDICIÓN DE TEXTO - BORRADO

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

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.

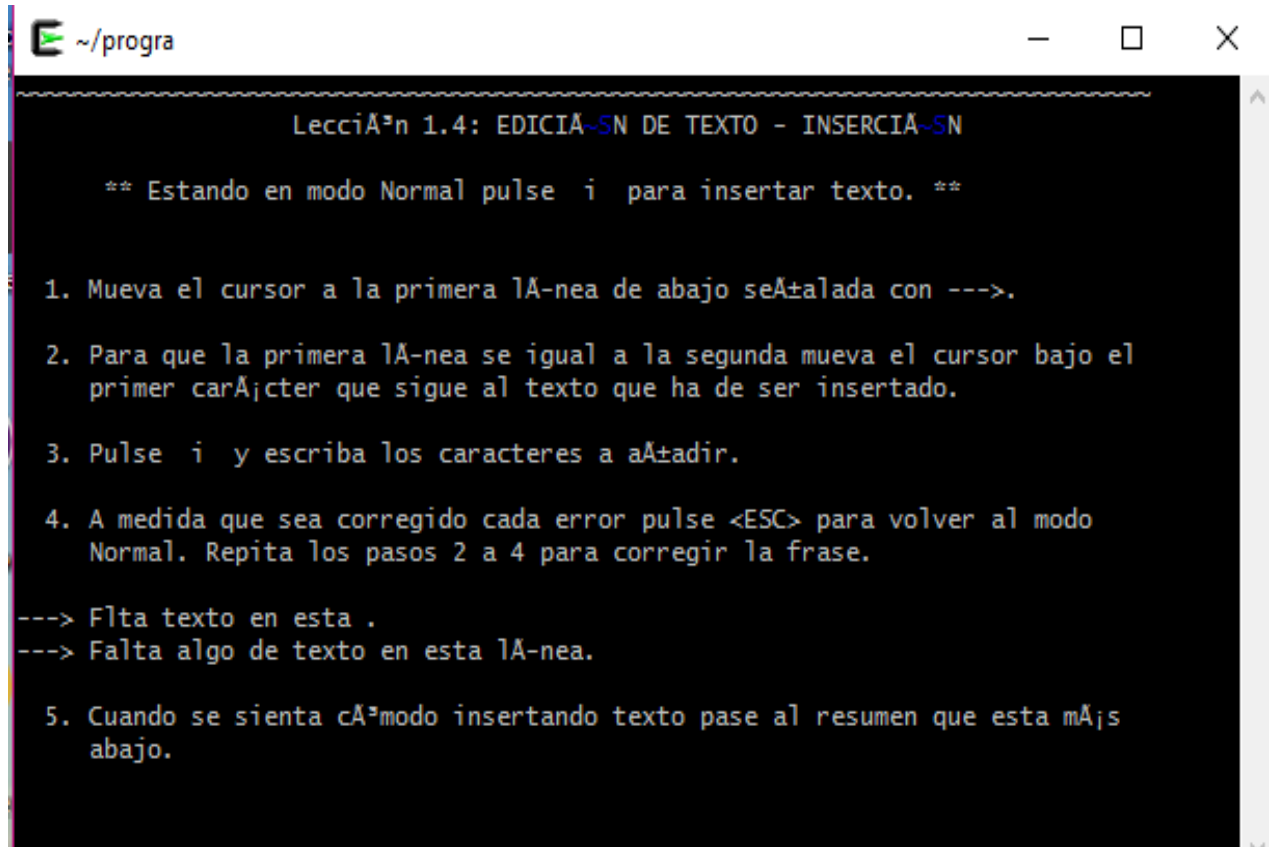
4. Repita los pasos 2 a 4 hasta que la frase sea la correcta.

---> 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.
```


Edición de texto a través del comando insertar "i".



```
~/progra
Lección 1.4: EDICIÓN DE TEXTO - INSERCIÓN

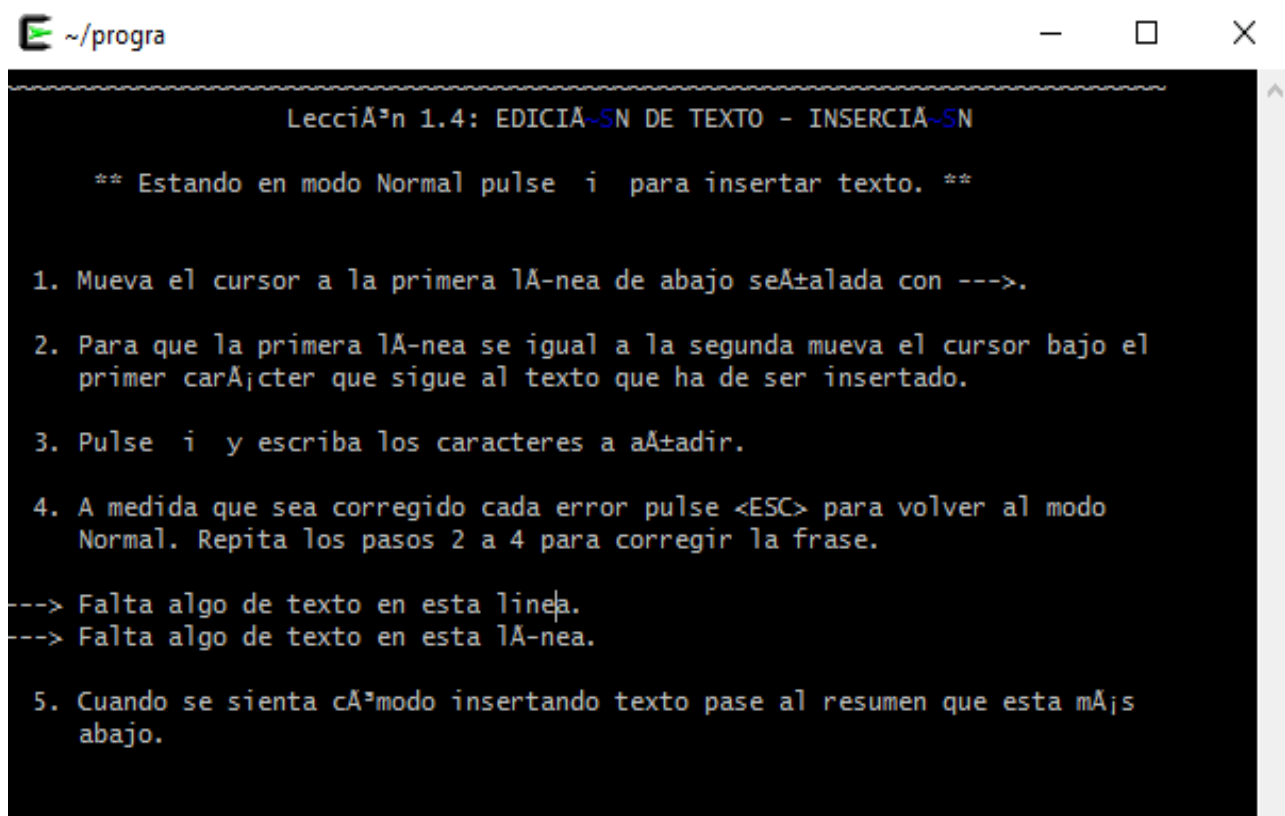
** Estando en modo Normal pulse i para insertar texto. **

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 texto en esta .
---> Falta algo de texto en esta línea.

5. Cuando se sienta cómodo insertando texto pase al resumen que está más
   abajo.
```

Insertando texto usando el comando aprendido.



```
~/progra
Lección 1.4: EDICIÓN DE TEXTO - INSERCIÓN

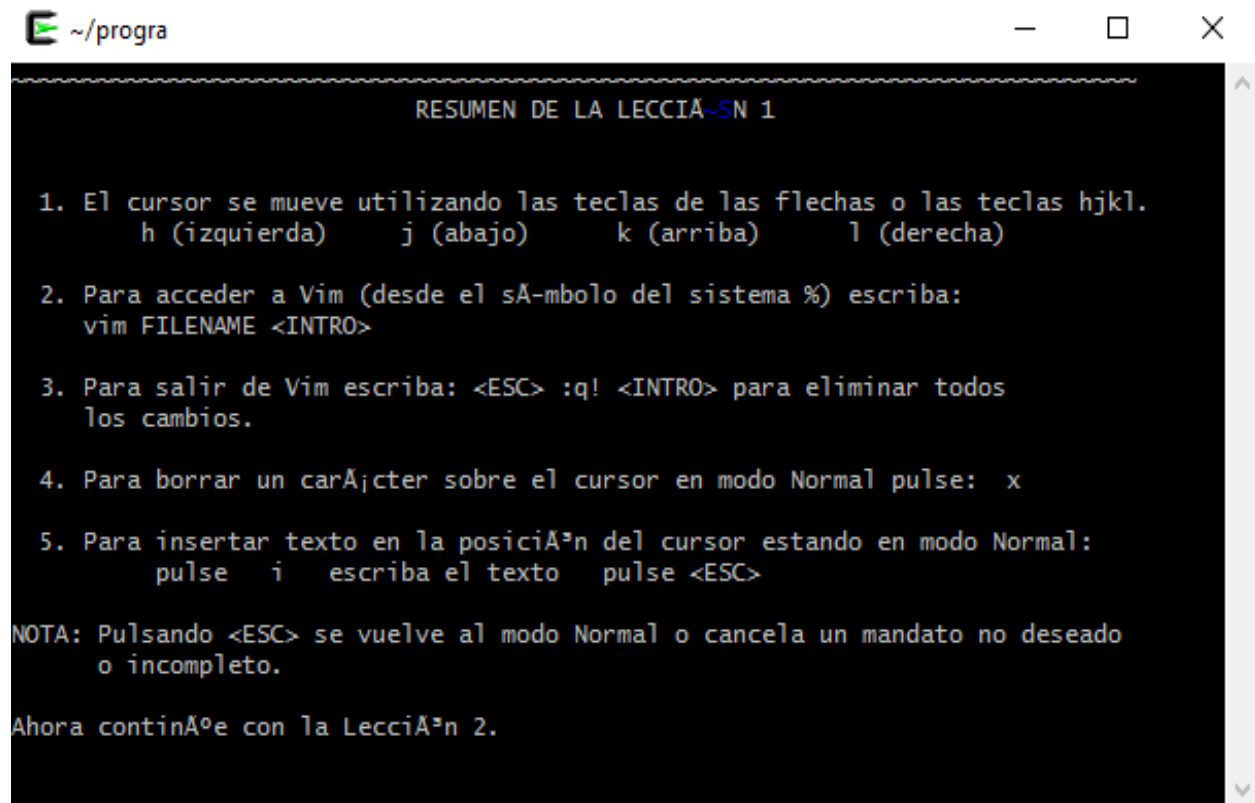
** Estando en modo Normal pulse i para insertar texto. **

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 está más
   abajo.
```

Resumen de la lección



```
~/progra

RESUMEN DE LA LECCIÓN 1

1. El cursor se mueve utilizando las teclas de las flechas o las teclas hjkl.
   h (izquierda)   j (abajo)   k (arriba)   l (derecha)

2. Para acceder a Vim (desde el símbolo del sistema %) escriba:
   vim FILENAME <INTRO>

3. Para salir de Vim escriba: <ESC> :q! <INTRO> para eliminar todos
   los cambios.

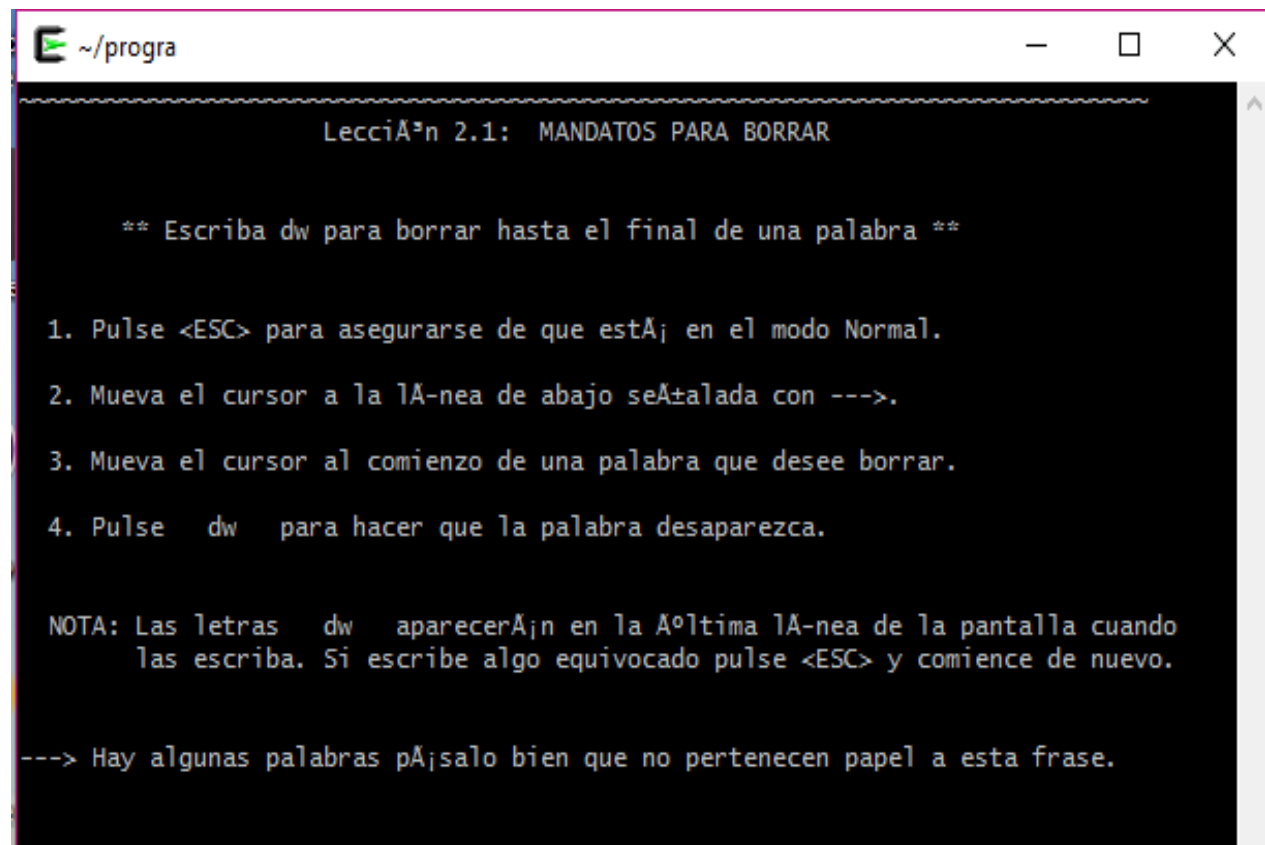
4. Para borrar un carácter sobre el cursor en modo Normal pulse: x

5. Para insertar texto en la posición del cursor estando en modo Normal:
   pulse i escriba el texto pulse <ESC>

NOTA: Pulsando <ESC> se vuelve al modo Normal o cancela un mandato no deseado
      o incompleto.

Ahora continúe con la Lección 2.
```

Aprendiendo a borrar palabras completas a través de mandatos (dw).



```
~/progra

Lección 2.1: MANDATOS PARA BORRAR

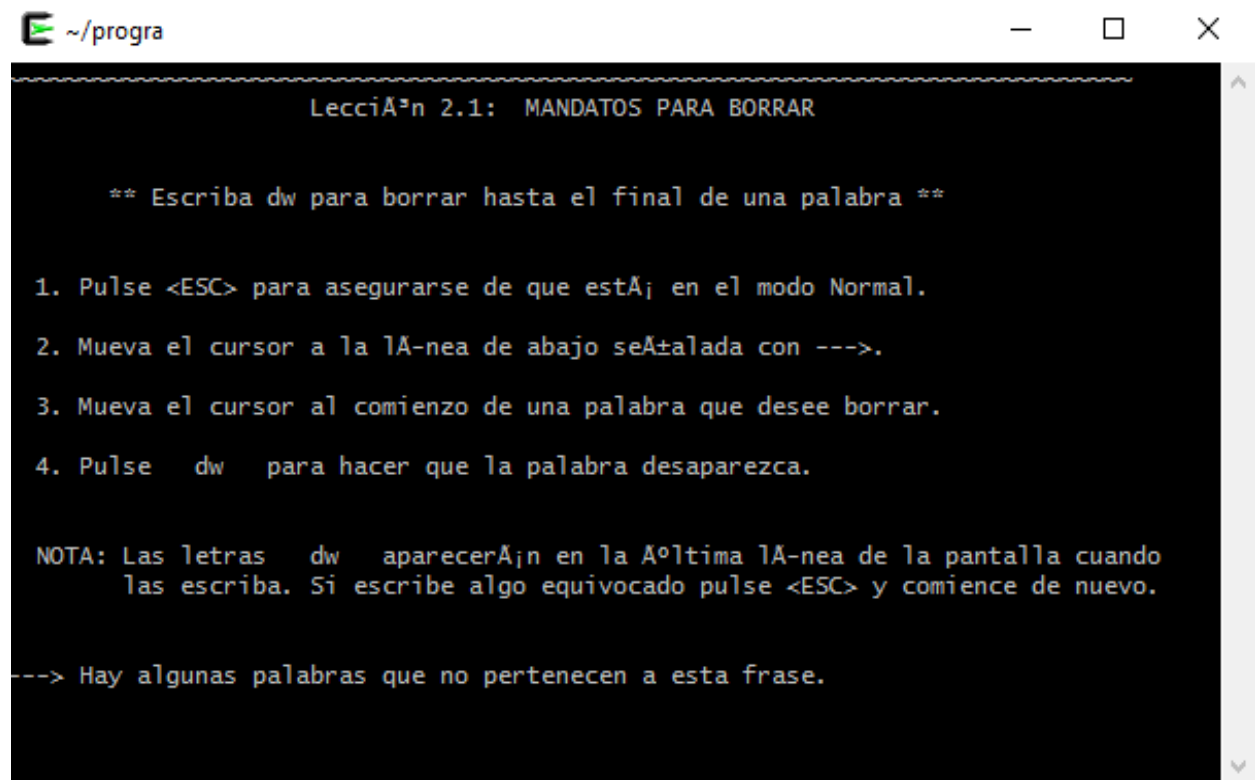
** Escriba dw para borrar hasta el final de una palabra **

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 más o menos que no pertenecen a esta frase.
```

Practica realizada



```
Lección 2.1: MANDATOS PARA BORRAR

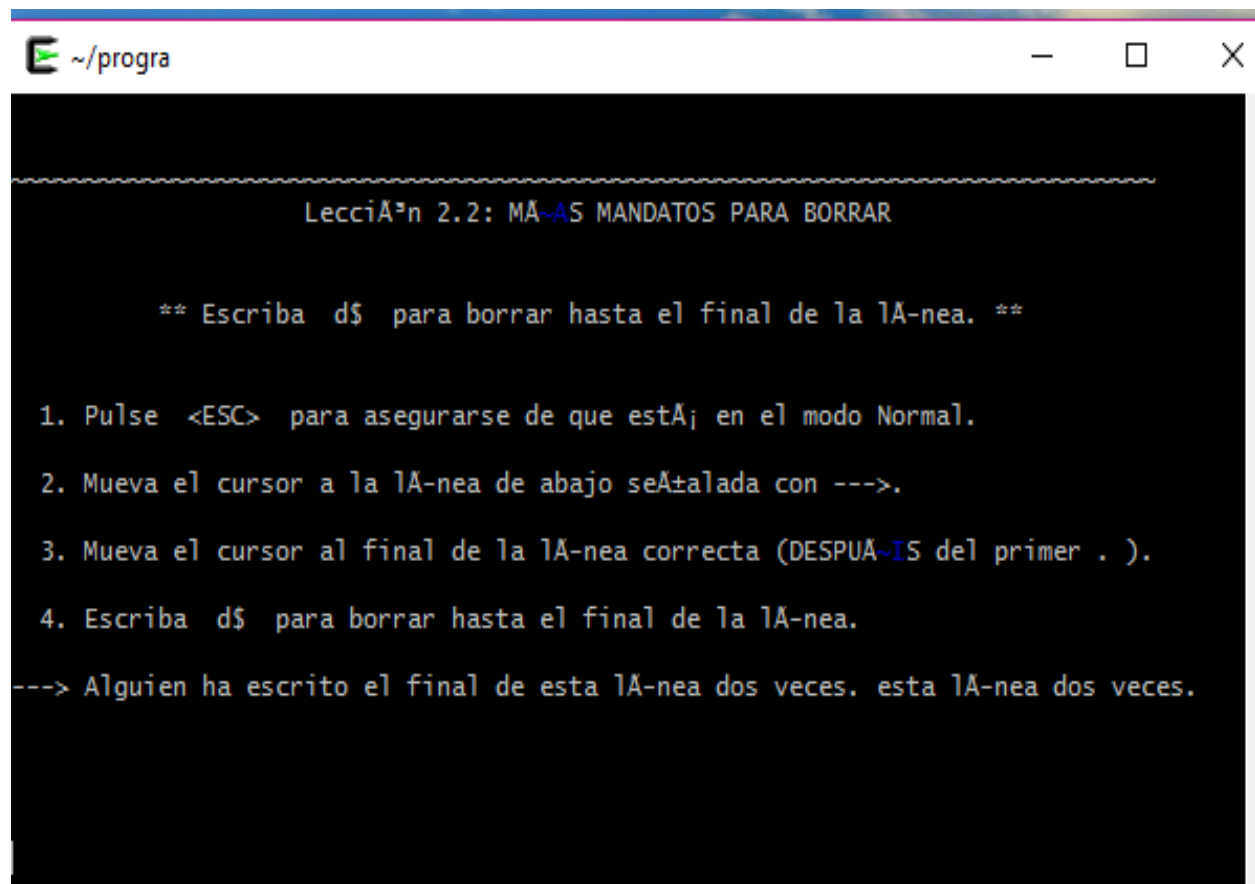
** Escriba dw para borrar hasta el final de una palabra **

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.
```

Utilización del comando d\$ para borrar una línea completa.



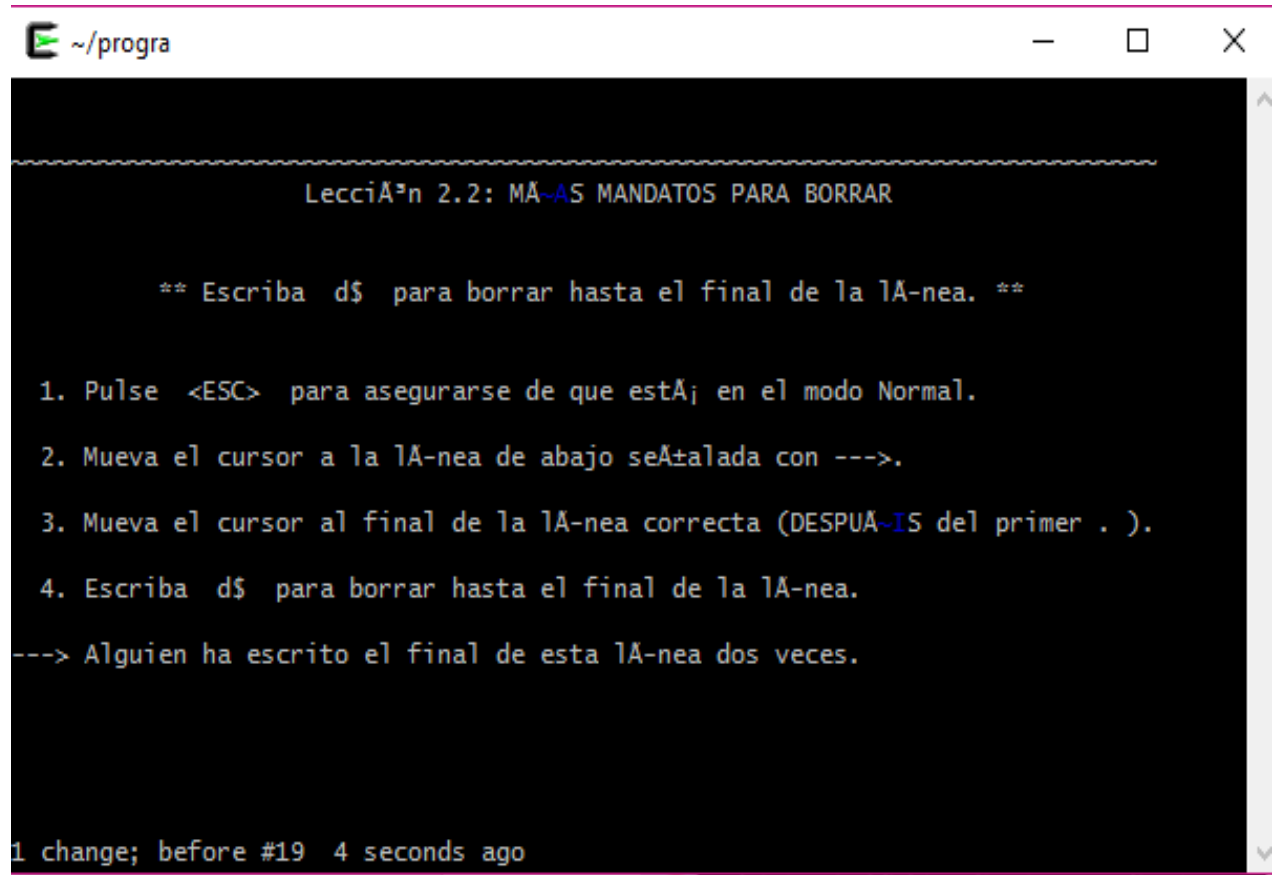
```
Lección 2.2: MÁS MANDATOS PARA BORRAR

** 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. esta línea dos veces.
```

Borrando la línea de sobra



```
~/progra

Lección 2.2: MÁS MANDATOS PARA BORRAR

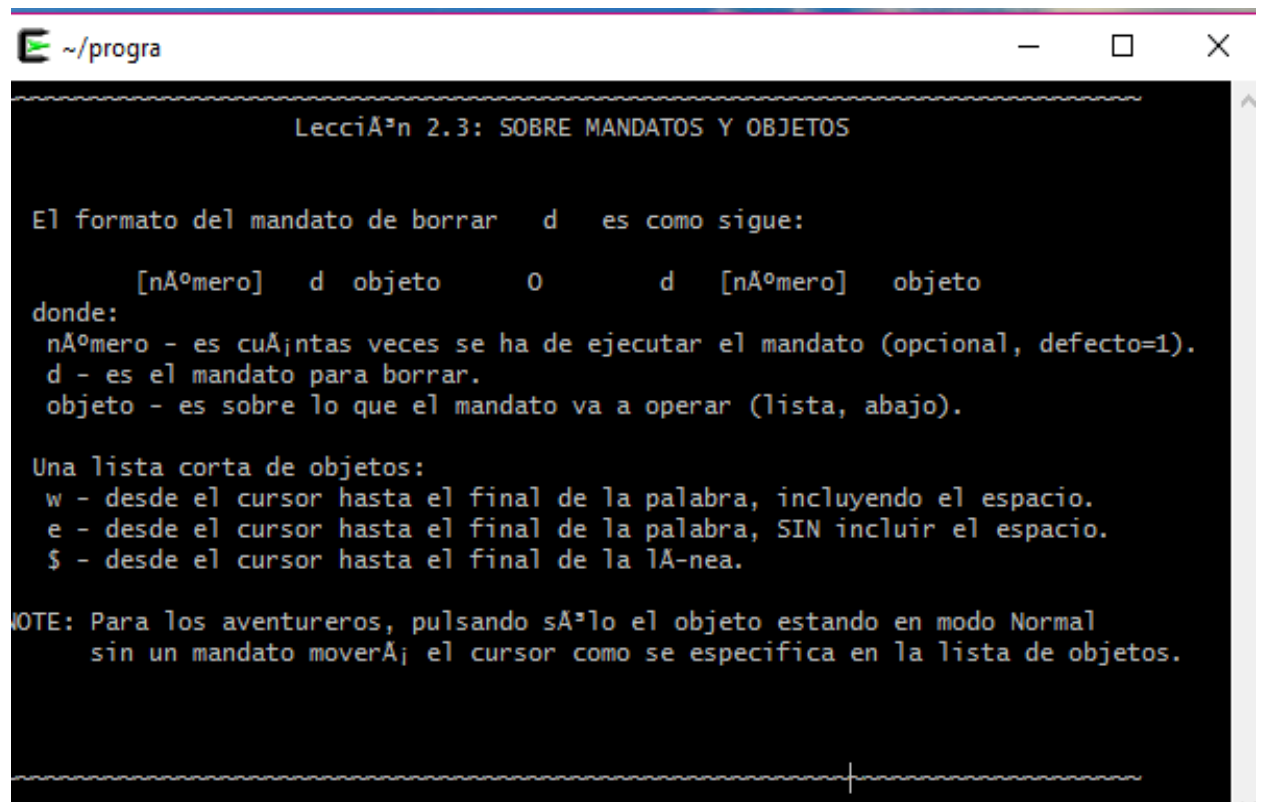
** Escriba d$ para borrar hasta el final de la línea. **

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.

1 change; before #19 4 seconds ago
```

Diferentes mandatos para borrar



```
~/progra

Lección 2.3: SOBRE MANDATOS Y OBJETOS

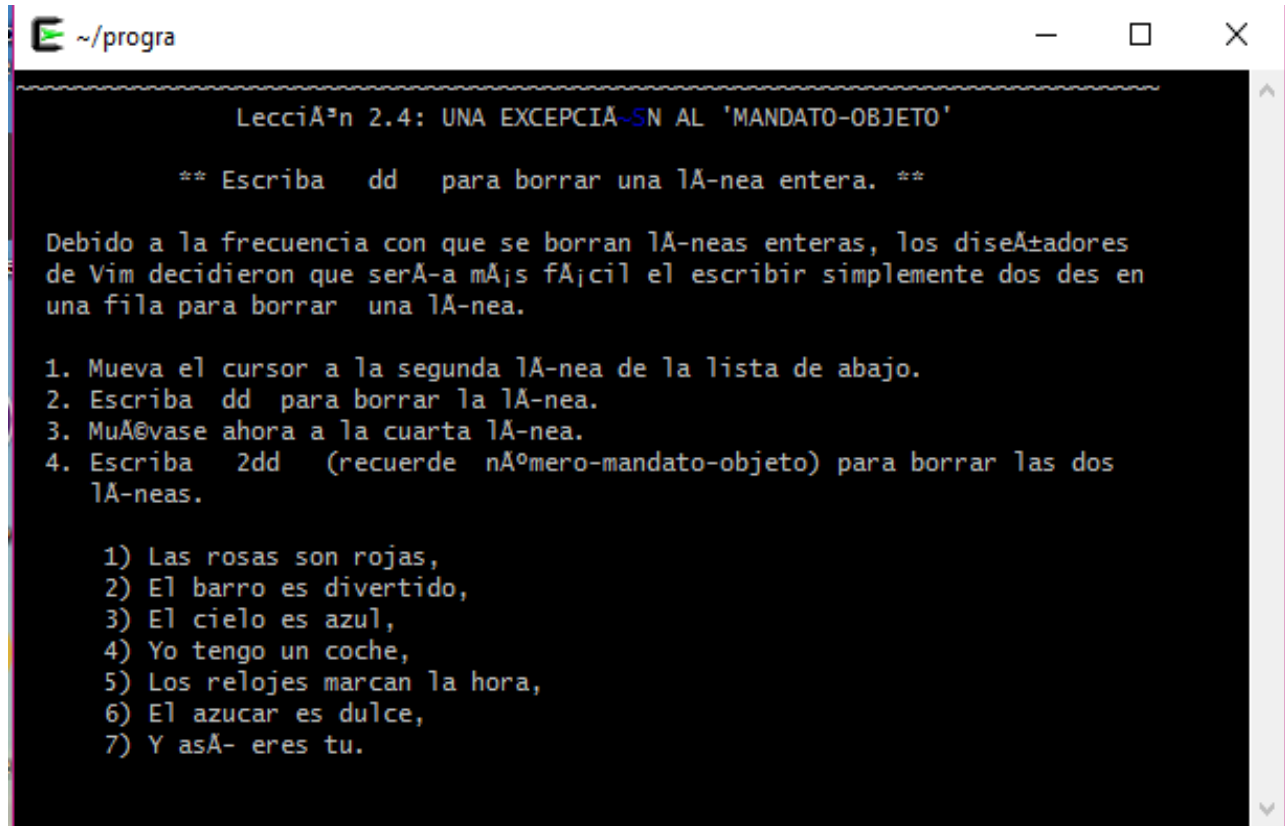
El formato del mandato de borrar d es como sigue:

[número] d objeto o d [número] objeto
donde:
número - es cuántas veces se ha de ejecutar el mandato (opcional, defecto=1).
d - es el mandato para borrar.
objeto - es sobre lo que el mandato va a operar (lista, abajo).

Una lista corta de objetos:
w - desde el cursor hasta el final de la palabra, incluyendo el espacio.
e - desde el cursor hasta el final de la palabra, SIN incluir el espacio.
$ - desde el cursor hasta el final de la línea.

NOTE: Para los aventureros, pulsando sólo el objeto estando en modo Normal
sin un mandato moverá el cursor como se especifica en la lista de objetos.
```

Borrando una línea completa utilizando “dd”.



```
~/progra

Lección 2.4: UNA EXCEPCIÓN AL 'MANDATO-OBJETO'

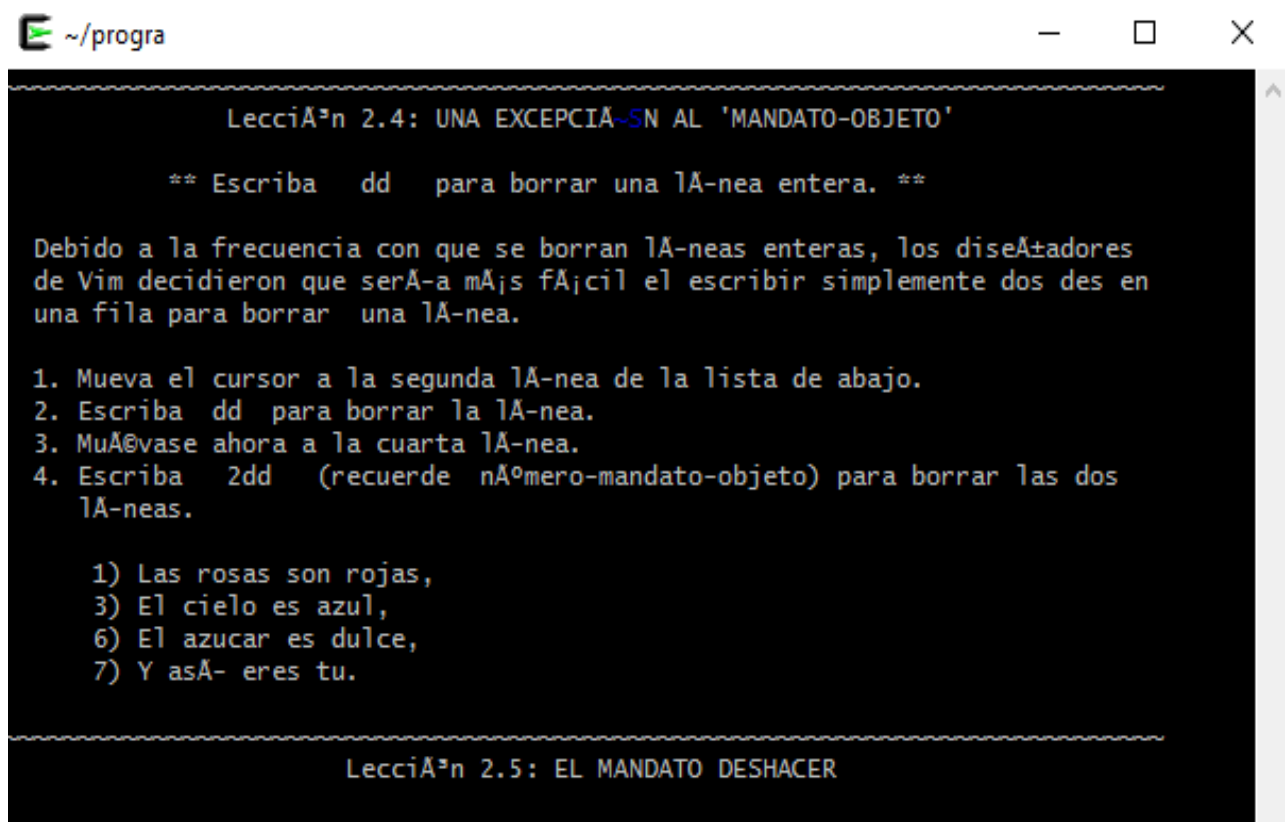
** Escriba dd para borrar una línea entera. **

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,
2) El barro es divertido,
3) El cielo es azul,
4) Yo tengo un coche,
5) Los relojes marcan la hora,
6) El azúcar es dulce,
7) Y así- eres tu.
```

Aplicando lo aprendido.



```
~/progra

Lección 2.4: UNA EXCEPCIÓN AL 'MANDATO-OBJETO'

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

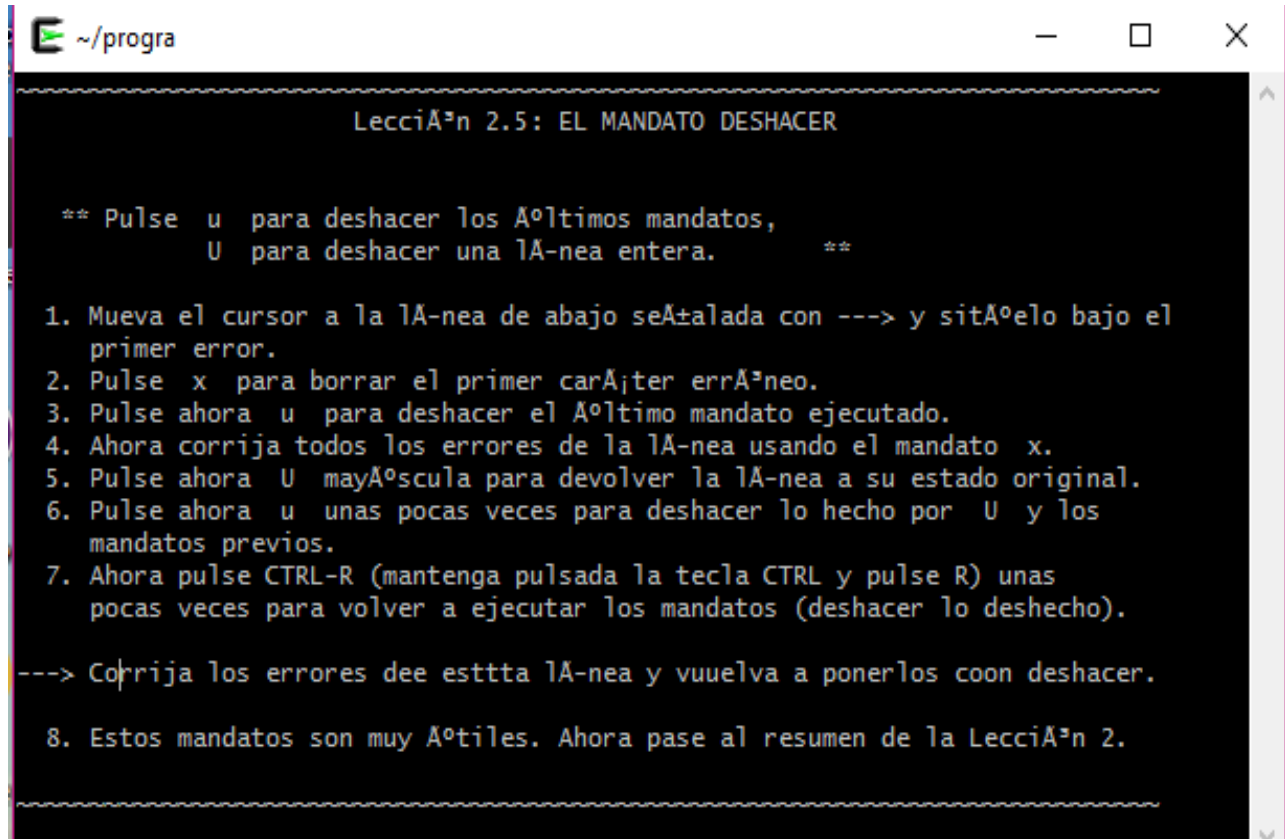
Debido a la frecuencia con que se borran líneas enteras, los diseñadores
de Vim decidieron que sería más fácil el escribir simplemente dos des en
una fila para borrar una línea.

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.

Lección 2.5: EL MANDATO DESHACER
```

Mandatos para deshacer “u”, “U” y “CTRL-R”



```
~/progra

Lección 2.5: EL MANDATO DESHACER

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

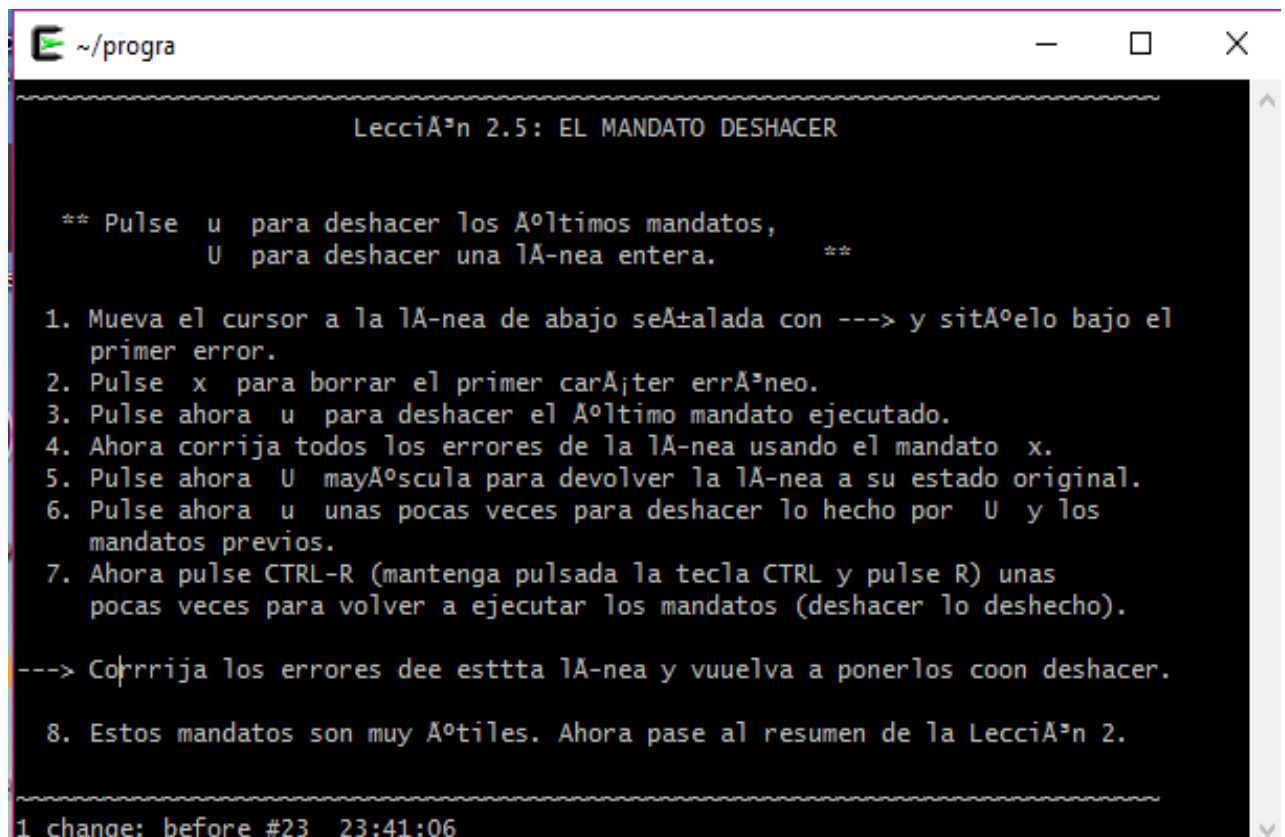
1. Mueva el cursor a la línea de abajo señalada con ---> y sitúelo bajo el
   primer error.
2. Pulse  x  para borrar el primer carácter erróneo.
3. Pulse ahora  u  para deshacer el último mandato ejecutado.
4. Ahora corrija todos los errores de la línea usando el mandato  x.
5. Pulse ahora  U  mayúscula 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.


```

Utilización de los mandatos. Utilizando u en este caso.



```
~/progra

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.

1 change; before #23 23:41:06
```

Utilizando U

```
~/progra

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.

1 change; before #25 6 seconds ago
```

Utilizando CTRL-R

```
~/progra

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.

Already at newest change
```


Resumiendo lo aprendido.

```
~/progra

RESUMEN DE LA LECCIÓN 2

1. Para borrar desde el cursor hasta el final de una palabra pulse: dw
2. Para borrar desde el cursor hasta el final de una línea pulse: d$
3. Para borrar una línea enter pulse: dd
4. El formato de un mandato en modo Normal es:

    [número] mandato objeto O mandato [número] objeto
donde:
número - es cuántas veces se ha de ejecutar el mandato
mandato - es lo que hay que hacer, por ejemplo, d para borrar
objeto - es sobre lo que el mandato va a operar, por ejemplo
        w (palabra), $ (hasta el final de la línea), etc.

5. Para deshacer acciones previas pulse: u (u minúscula)
Para deshacer todos los cambios de una línea pulse: U (U mayúscula)
Para deshacer lo deshecho pulse: CTRL-R
```

Aprendiendo a poner.

```
~/progra

Lección 3.1: EL MANDATO A«PUTA» (poner)

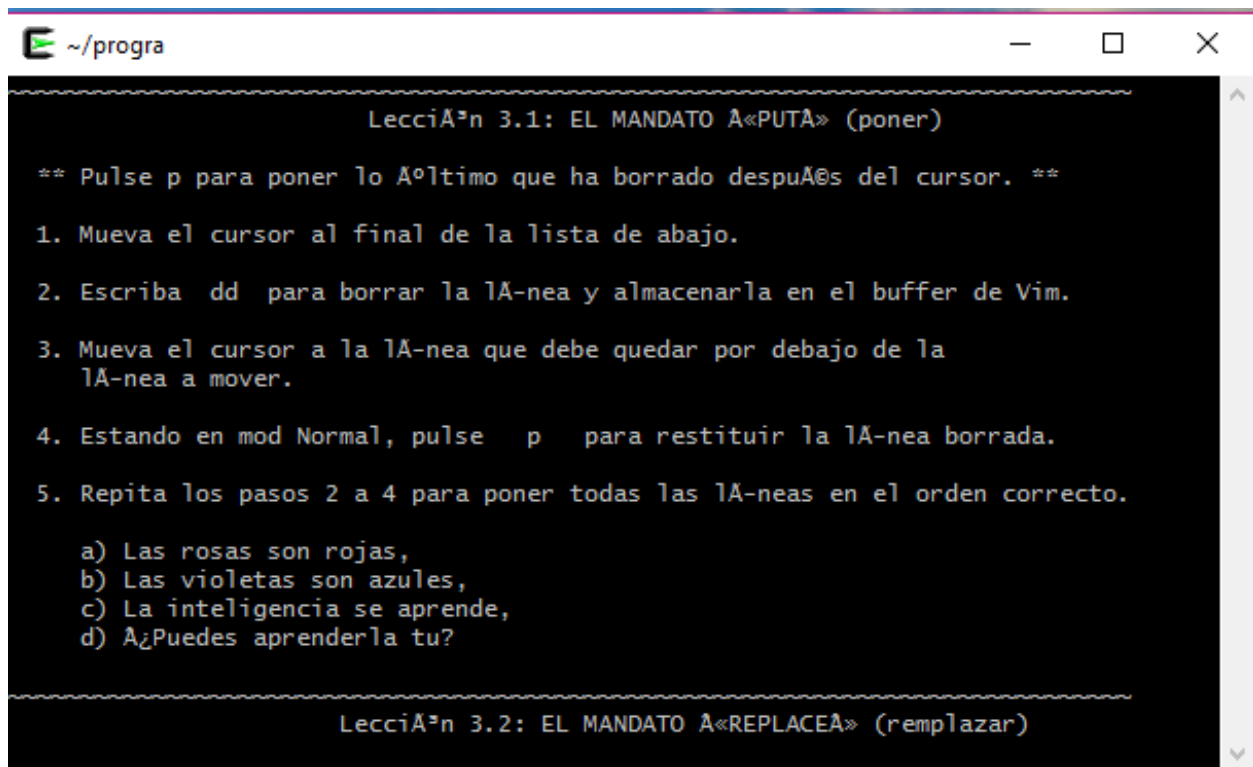
** Pulse p para poner lo último que ha borrado después del cursor. **

1. Mueva el cursor al final de la lista de abajo.
2. Escriba dd para borrar la línea y almacenarla en el buffer de Vim.
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.

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

Lección 3.2: EL MANDATO A«REPLACEA» (reemplazar)
```


Utilizando “dd” para borrar y guardar la información en el bufer y después utilizando “p” para colocarlo donde se requiere.



```
~/progra

Lección 3.1: EL MANDATO A«PUTA» (poner)

** Pulse p para poner lo Aºltimo que ha borrado despu s del cursor. **

1. Mueva el cursor al final de la lista de abajo.

2. Escriba dd para borrar la l nea y almacenarla en el buffer de Vim.

3. Mueva el cursor a la l nea que debe quedar por debajo de la
   l nea a mover.

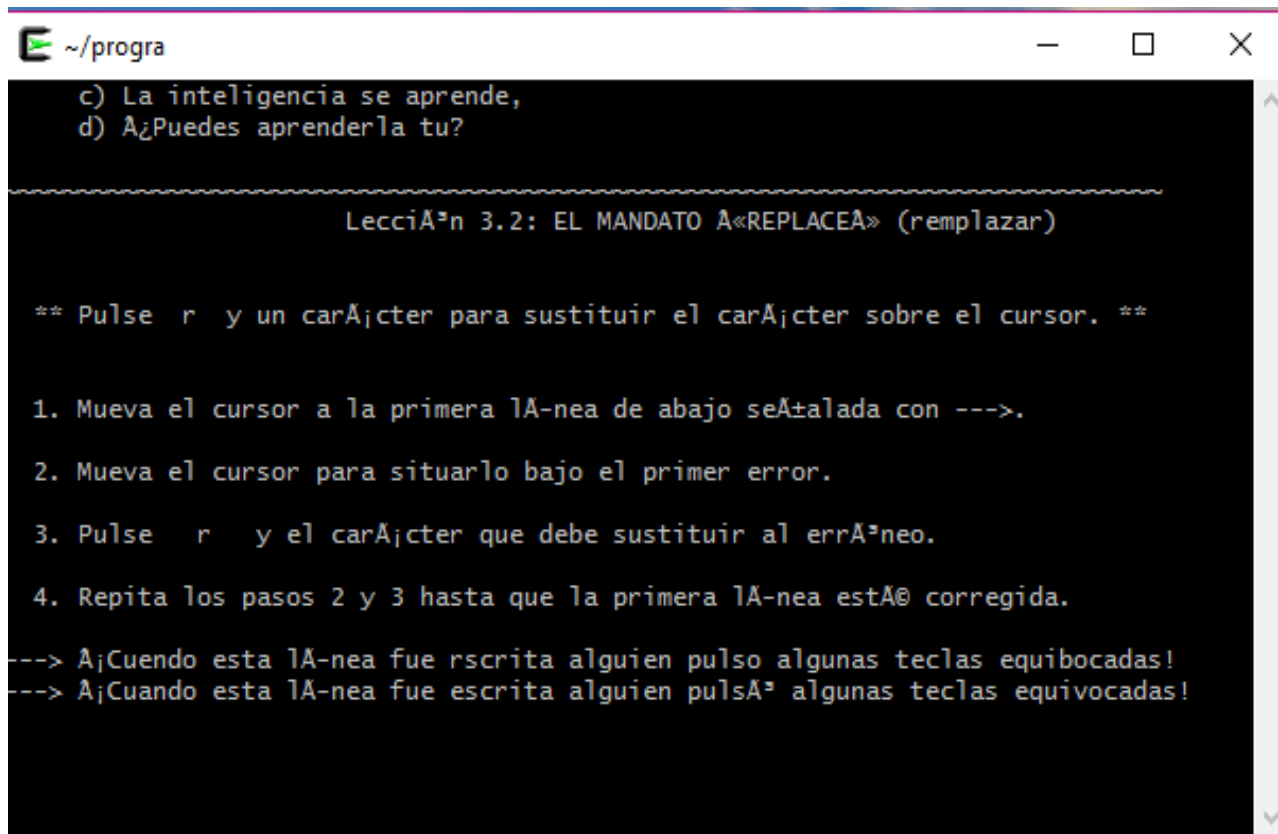
4. Estando en mod Normal, pulse p para restituir la l nea borrada.

5. Repita los pasos 2 a 4 para poner todas las l neas en el orden correcto.

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

Lección 3.2: EL MANDATO A«REPLACEA» (reemplazar)
```

Aprendiendo a reemplazar un car cter.



```
~/progra

c) La inteligencia se aprende,
d)  Puedes aprenderla tu?

Lección 3.2: EL MANDATO A«REPLACEA» (reemplazar)

** Pulse r y un car cter para sustituir el car cter sobre el cursor. **

1. Mueva el cursor a la primera l nea de abajo se alada con --->.

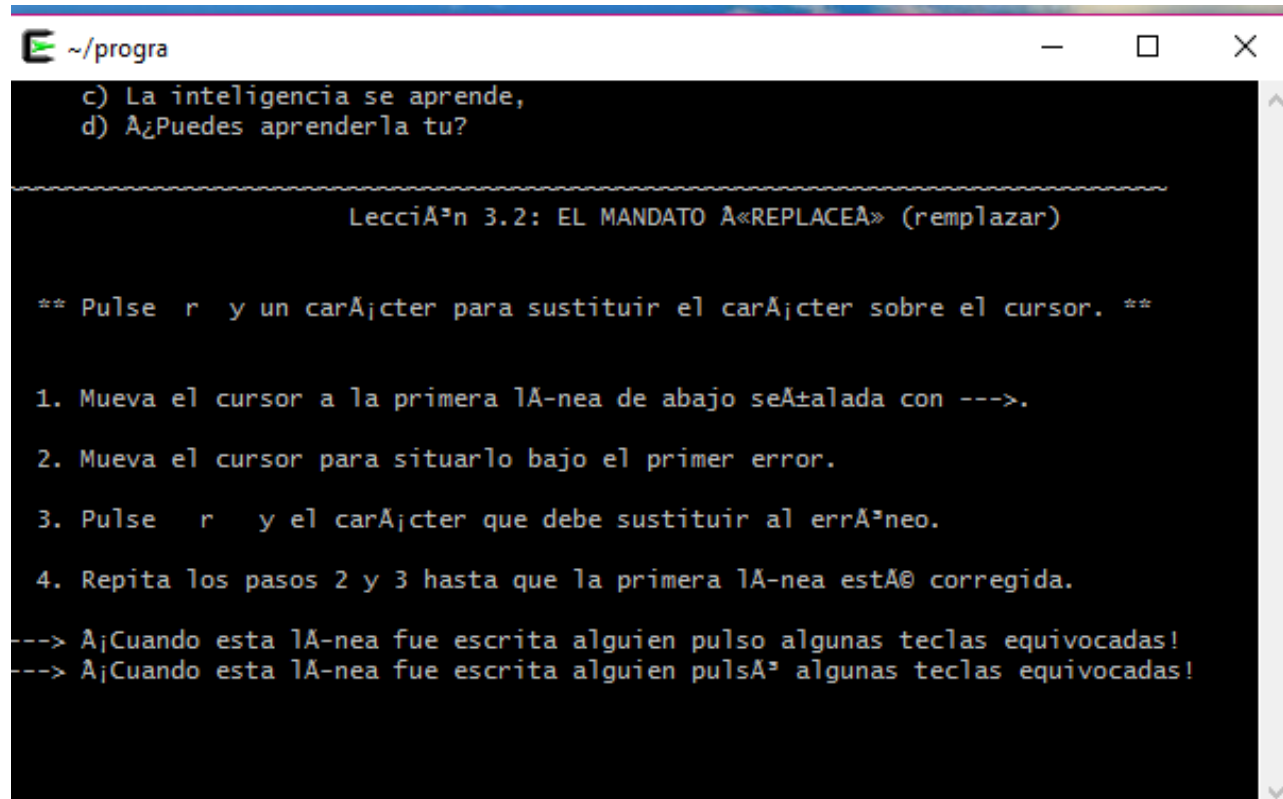
2. Mueva el cursor para situarlo bajo el primer error.

3. Pulse r y el car cter que debe sustituir al err neo.

4. Repita los pasos 2 y 3 hasta que la primera l nea est  corregida.

--->  Cuendo esta l nea fue rscrita alguien pulso algunas teclas equibocadas!
--->  Cuando esta l nea fue escrita alguien puls  algunas teclas equivocadas!
```

Utilizando “r” reemplazo los caracteres que están equivocados en una línea



```
~/progra
c) La inteligencia se aprende,
d) ¿Puedes aprenderla tu?

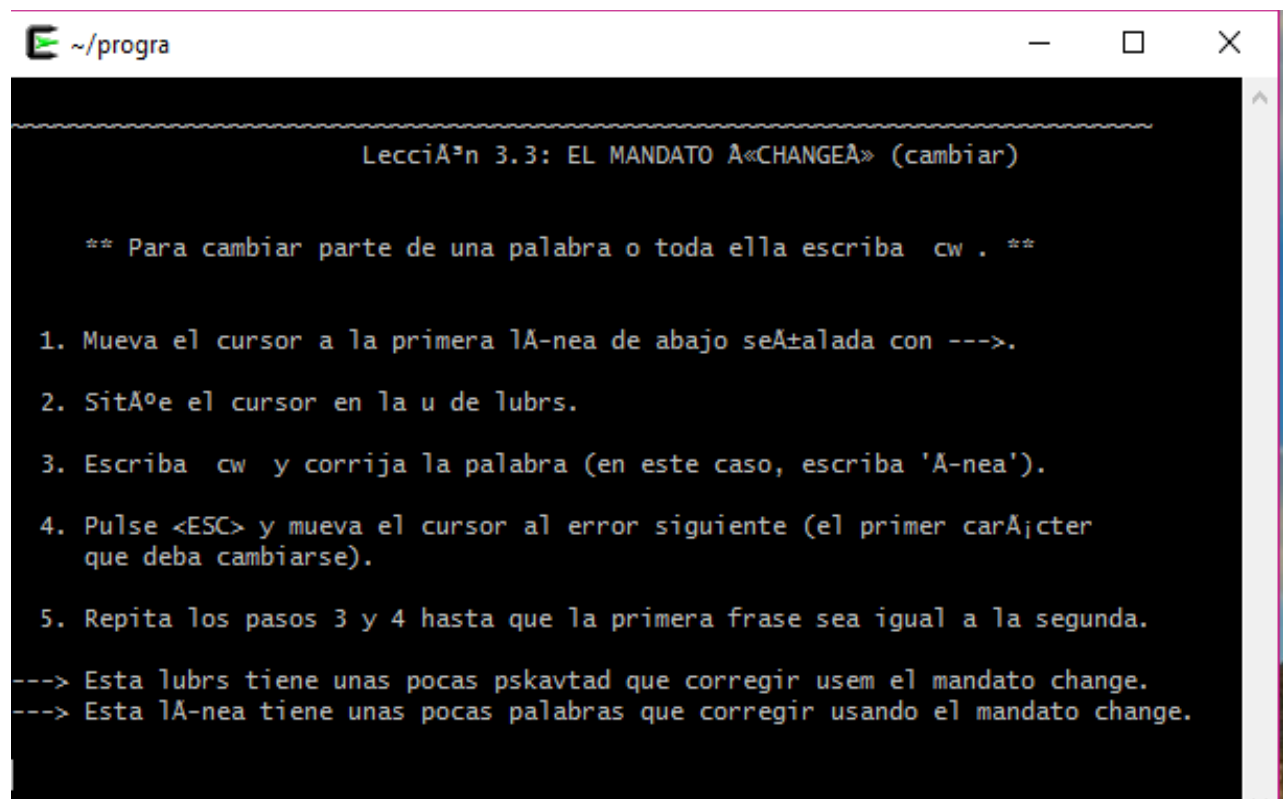
-----
Lección 3.2: EL MANDATO «REPLACEA» (reemplazar)

** 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 pulso algunas teclas equivocadas!
---> ¿Cuando esta línea fue escrita alguien pulsó algunas teclas equivocadas!
```

Aprendiendo el comando de cambiar “cw”, lo cual puede cambiar parte de una palabra o una palabra completa.



```
~/progra

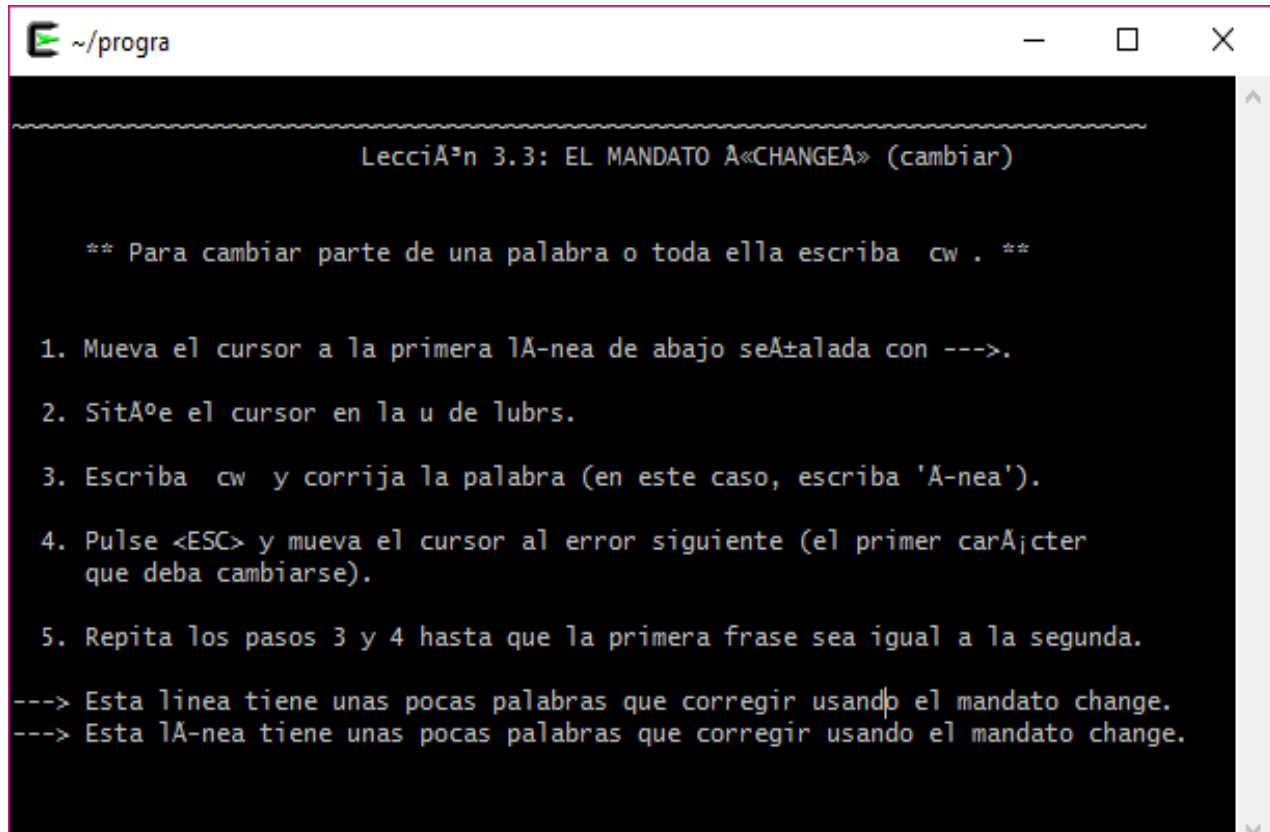
-----
Lección 3.3: EL MANDATO «CHANGEA» (cambiar)

** 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 'línea').
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 lubrs tiene unas pocas pskavtd que corregir usem el mandato change.
---> Esta línea tiene unas pocas palabras que corregir usando el mandato change.
```

Haciendo el ejercicio con el comando aprendido.



A terminal window titled "~/progra" with standard window controls. The content is a lesson titled "Lección 3.3: EL MANDATO A«CHANGEA» (cambiar)". It explains the 'cw' command for changing words. The text is as follows:

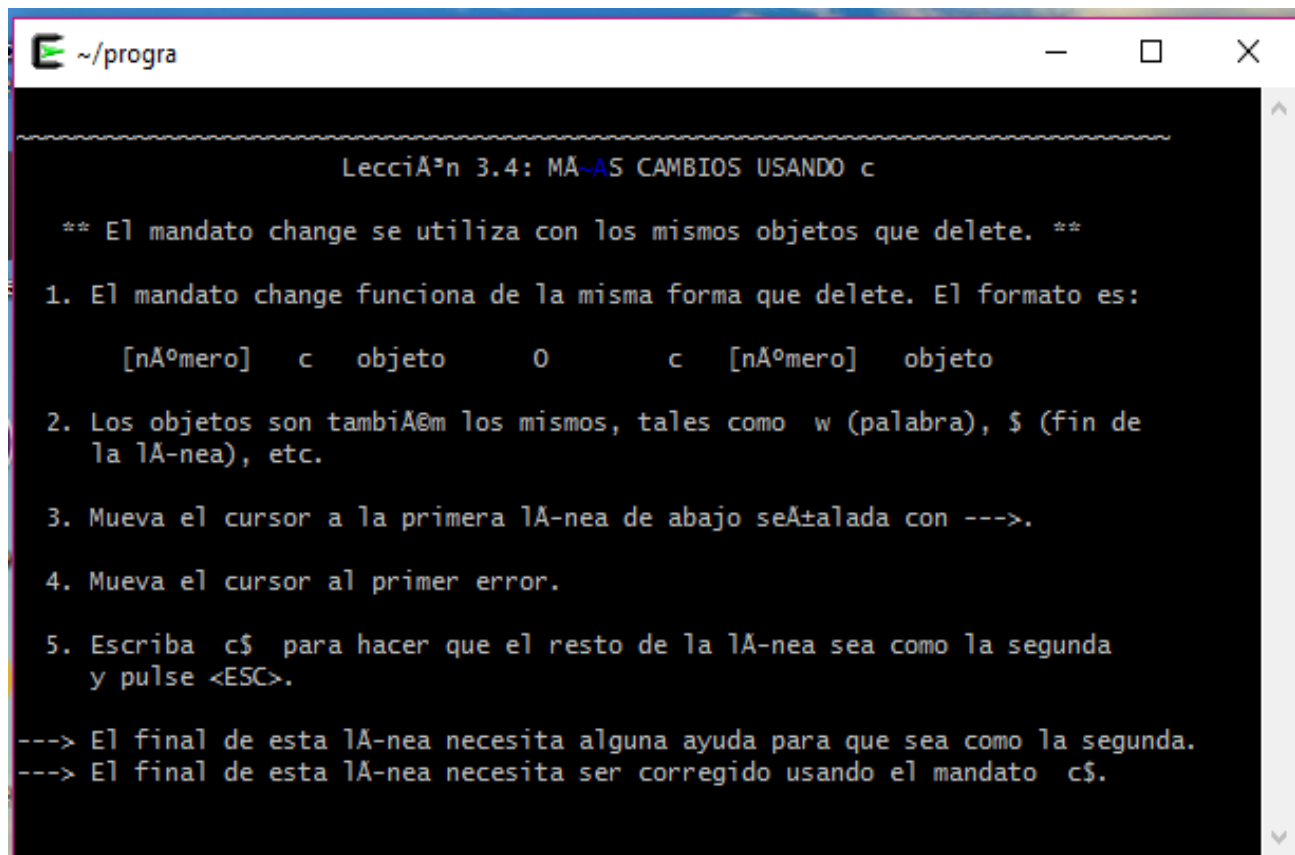
```
~~~~~
Lección 3.3: EL MANDATO A«CHANGEA» (cambiar)

** 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 'A-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.
```

Mandato “c\$” para cambiar más de una palabra dentro de una línea.



A terminal window titled "~/progra" with standard window controls. The content is a lesson titled "Lección 3.4: MÁS CAMBIOS USANDO c". It explains the 'c\$' command for changing multiple words in a line. The text is as follows:

```
~~~~~
Lección 3.4: MÁS 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:

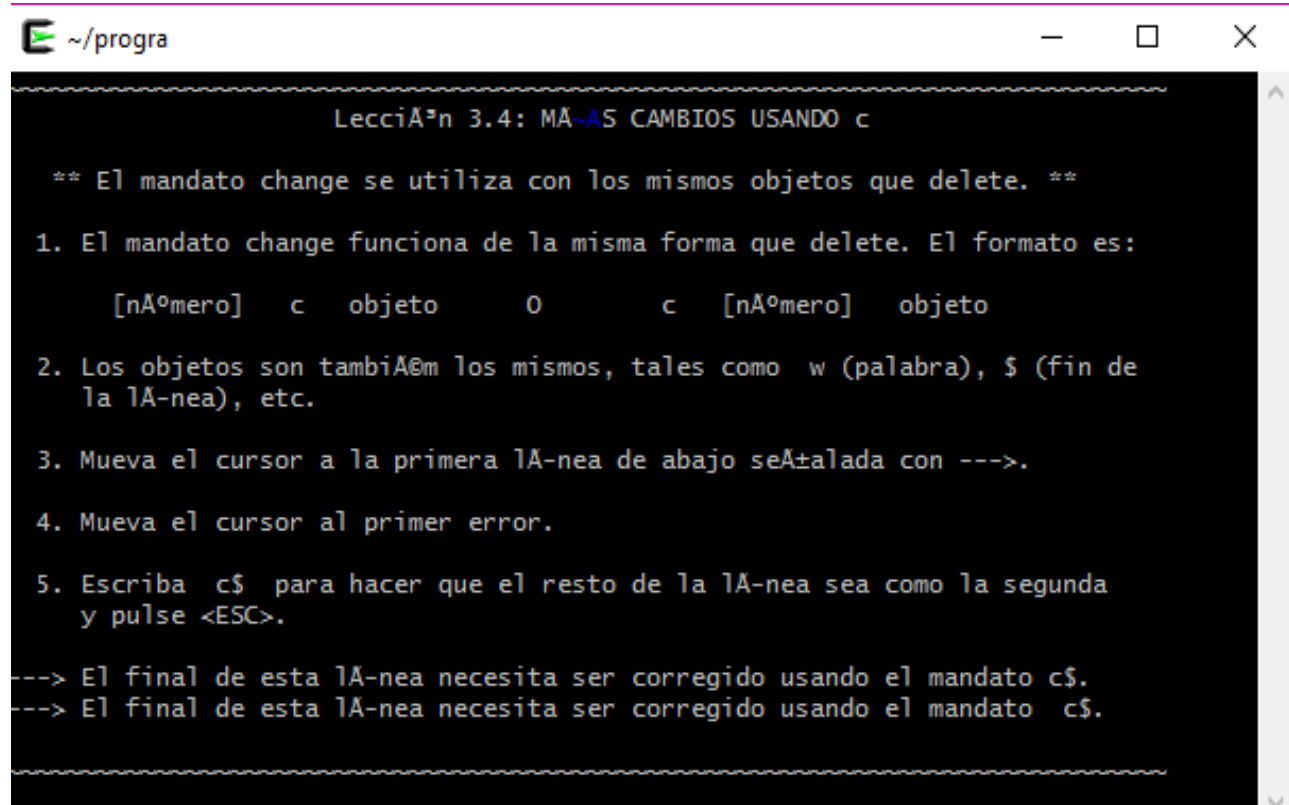
    [número] c objeto o c [número] objeto

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 alguna ayuda para que sea como la segunda.
---> El final de esta línea necesita ser corregido usando el mandato c$.
```

Aplicando lo aprendido.



```
~ /progra

Lección 3.4: MÁS 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.

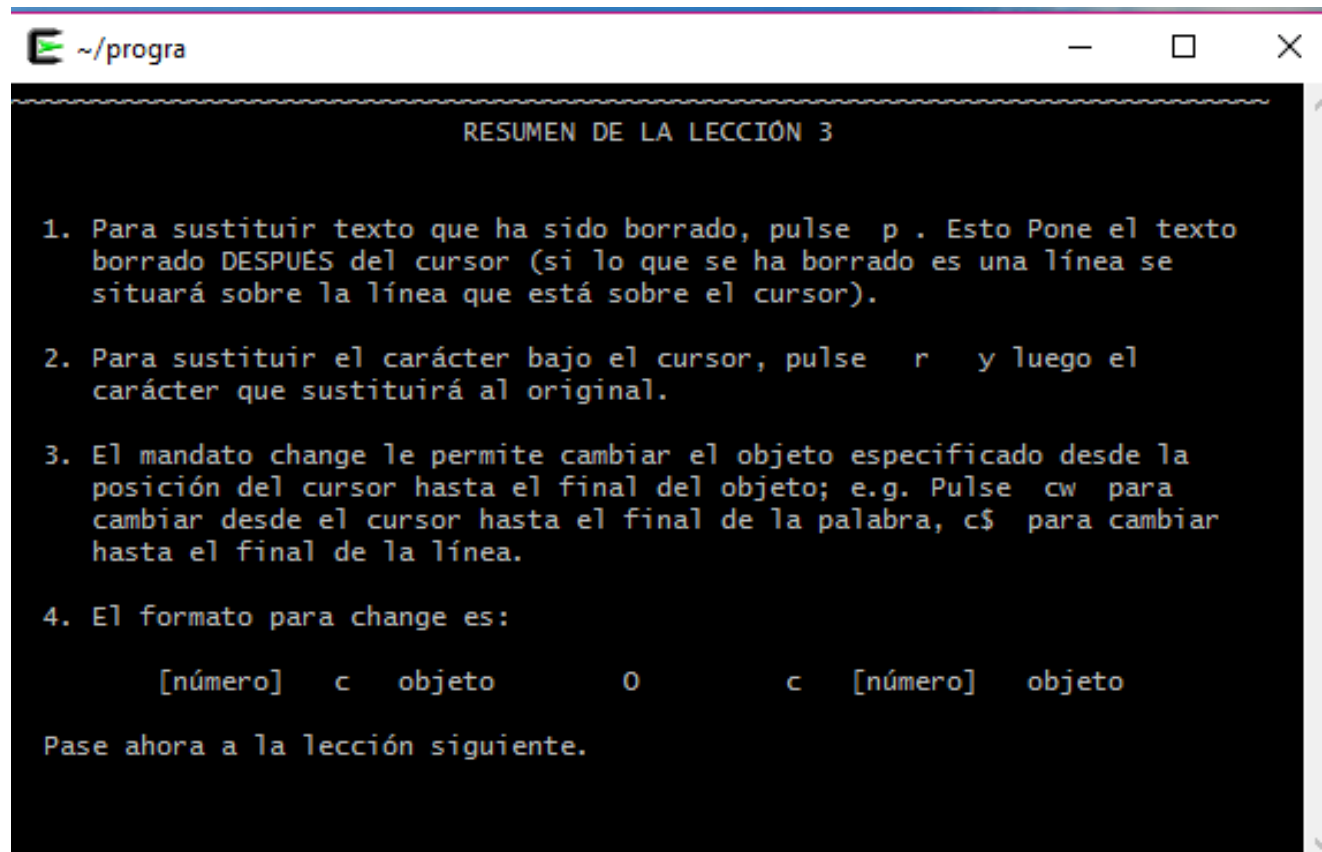
3. Mueva el cursor a la primera línea de abajo señalada con --->.

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$.
---> El final de esta línea necesita ser corregido usando el mandato c$.
```

Resumiendo lo aprendido en esta lección.



```
~ /progra

RESUMEN DE LA LECCIÓN 3

1. Para sustituir texto que ha sido borrado, pulse p . Esto pone el texto
   borrado DESPUÉS del cursor (si lo que se ha borrado es una línea se
   situará sobre la línea que está sobre el cursor).

2. Para sustituir el carácter bajo el cursor, pulse r y luego el
   carácter que sustituirá al original.

3. El mandato change le permite cambiar el objeto especificado desde la
   posición del cursor hasta el final del objeto; e.g. Pulse cw para
   cambiar desde el cursor hasta el final de la palabra, c$ para cambiar
   hasta el final de la línea.

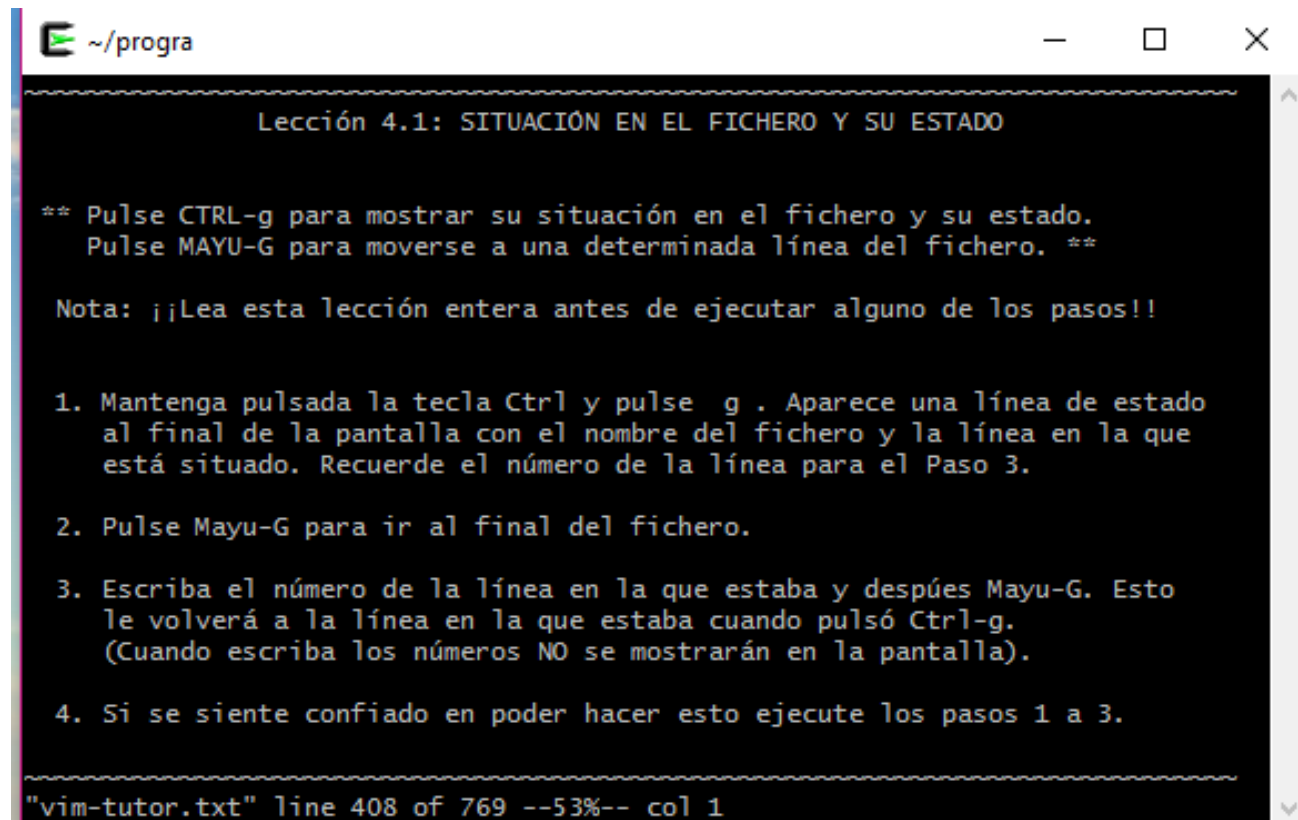
4. El formato para change es:

    [número] c objeto 0 c [número] objeto

Pase ahora a la lección siguiente.
```

Utilizando los mandatos CTRL-g y Mayu-G para saber el estado del fichero.

CTRL-g nos indicara en que línea del fichero nos encontramos y nos indicara en que estado esta.



```
~/progra
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.

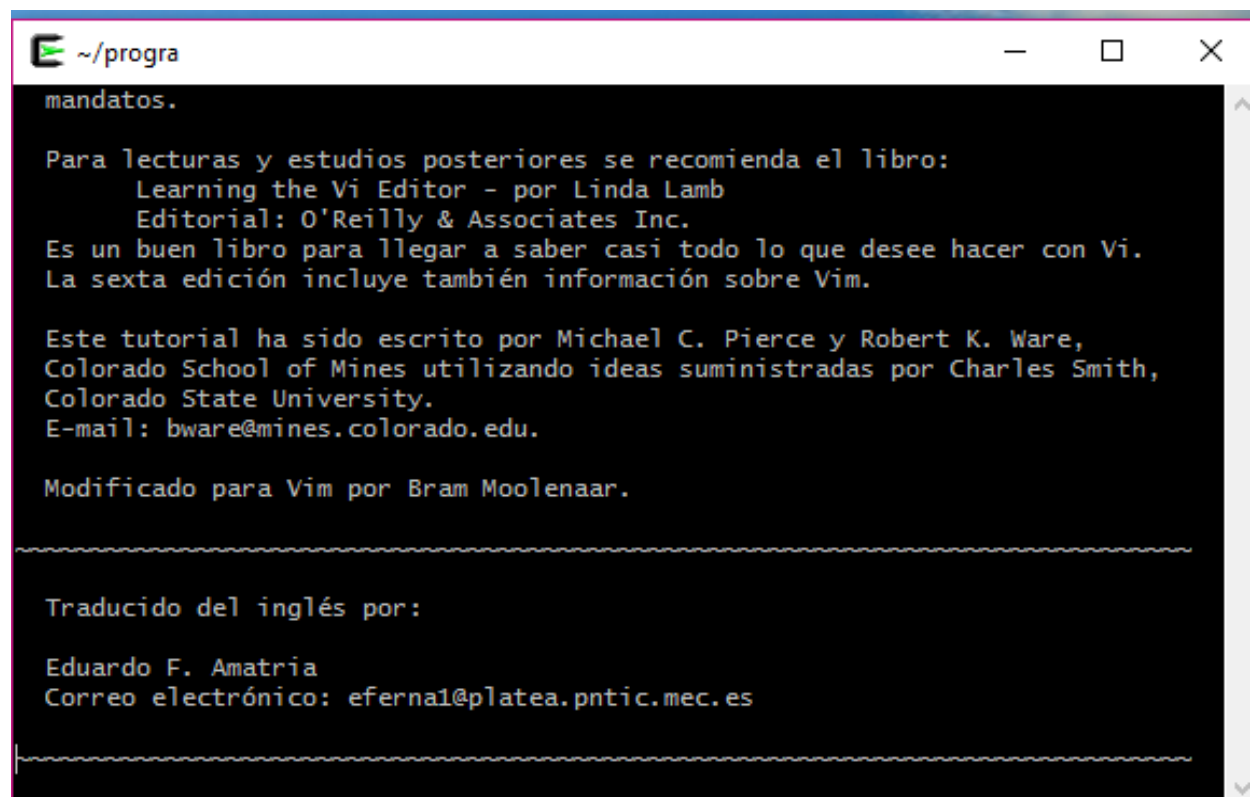
2. Pulse Mayu-G para ir al final del fichero.

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).

4. Si se siente confiado en poder hacer esto ejecute los pasos 1 a 3.

"vim-tutor.txt" line 408 of 769 --53%-- col 1
```

Mayu-G nos hará ir al final del fichero.



```
~/progra
mandatos.

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.

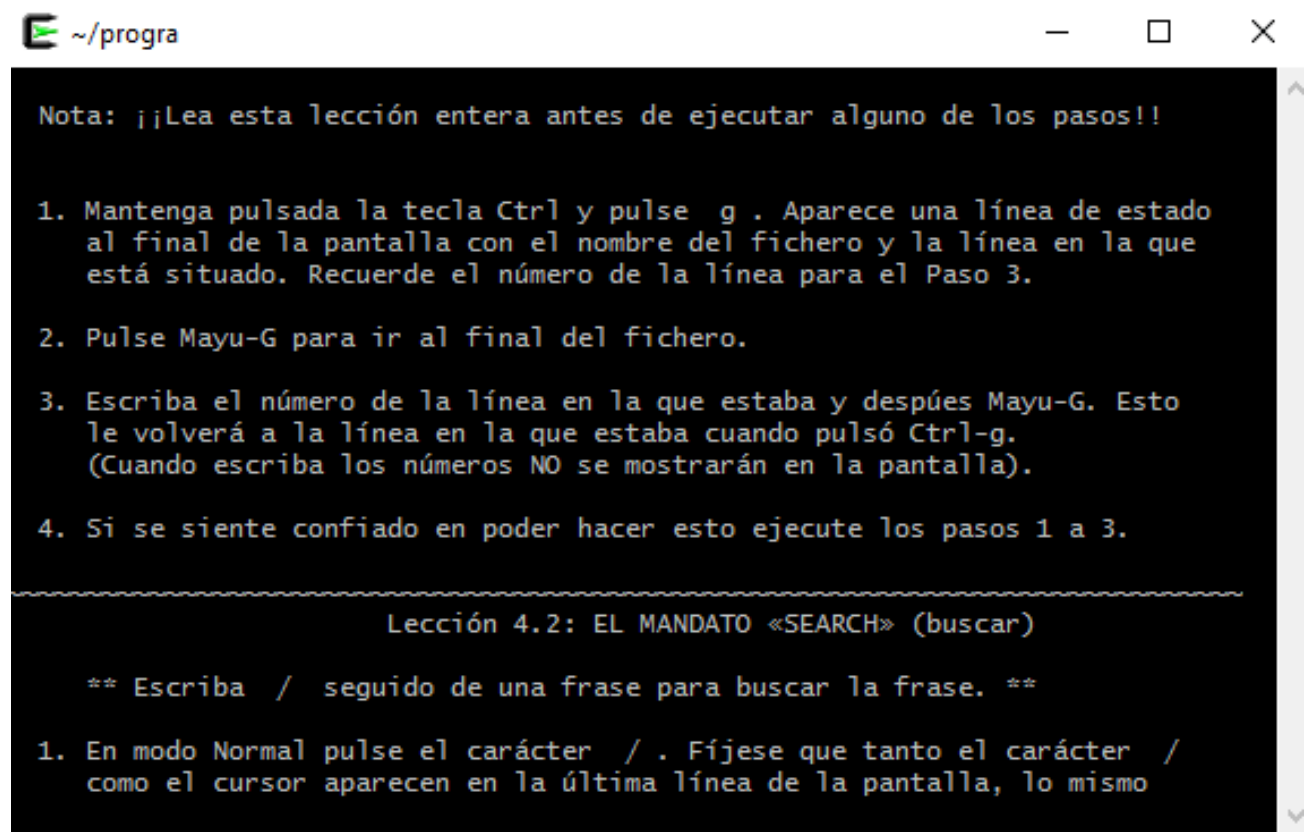
Este tutorial ha sido escrito por Michael C. Pierce y Robert K. Ware,
Colorado School of Mines utilizando ideas suministradas por Charles Smith,
Colorado State University.
E-mail: bware@mines.colorado.edu.

Modificado para Vim por Bram Moolenaar.

Traducido del inglés por:

Eduardo F. Amatria
Correo electrónico: eferna1@platea.pntic.mec.es
```

Para localizarnos a donde nos encontrábamos la última vez basta poner el numero de la línea y MAYU-G.



```
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).

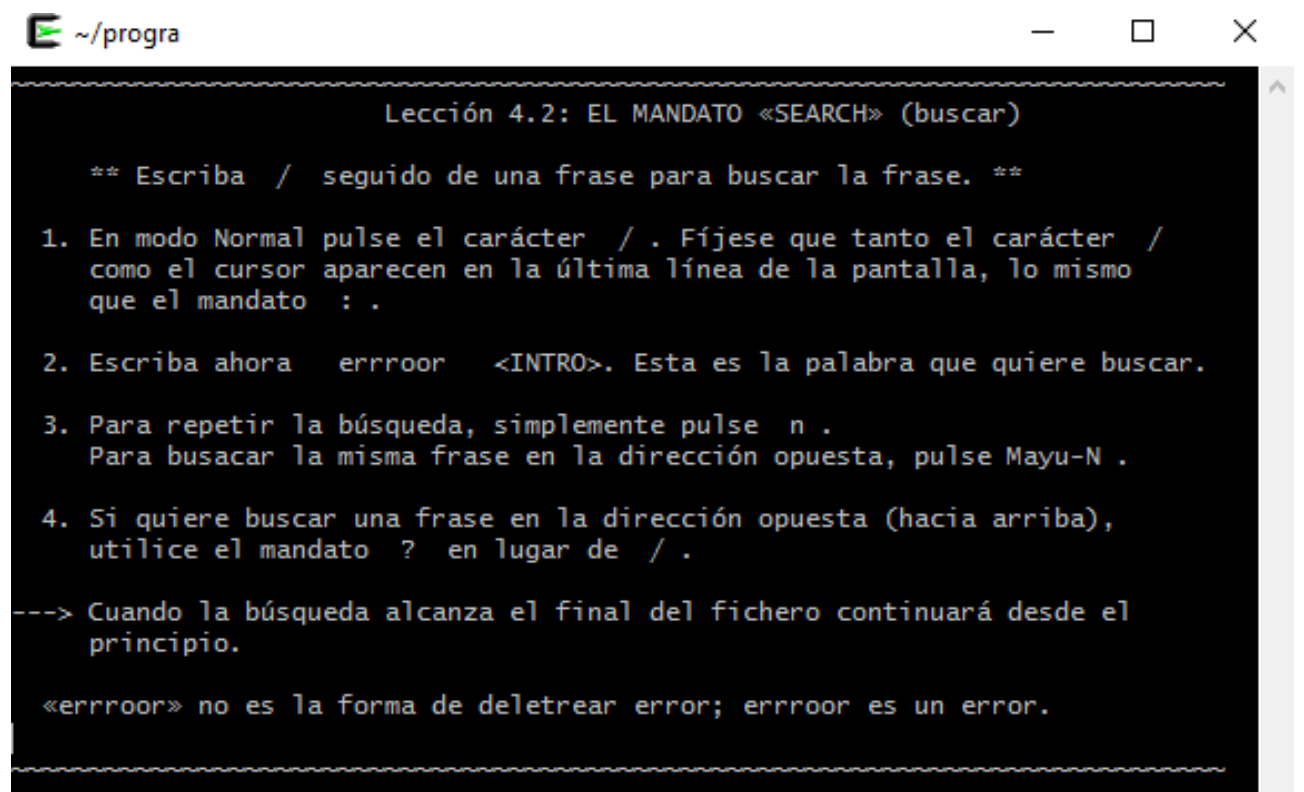
4. Si se siente confiado en poder hacer esto ejecute los pasos 1 a 3.

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

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

1. En modo Normal pulse el carácter / . Fíjese que tanto el carácter /
  como el cursor aparecen en la última línea de la pantalla, lo mismo
```

Mandato de Buscar (Search). Utilizando / frase que se quiere buscar.



```
-----
Lección 4.2: EL MANDATO «SEARCH» (buscar)

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

1. En modo Normal pulse el carácter / . Fíjese 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 busacar 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.

-----
```

Utilizando este mandato se hará más fácil encontrar algo en específico.

```
~/progra

Lección 4.2: EL MANDATO «SEARCH» (buscar)

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

1. En modo Normal pulse el carácter /. Fíjese 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 busacar 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.

/errroor|
```

```
~/progra

Lección 4.2: EL MANDATO «SEARCH» (buscar)

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

1. En modo Normal pulse el carácter /. Fíjese 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 busacar la misma frase en la dirección opuesta, pulse Mayu-N .

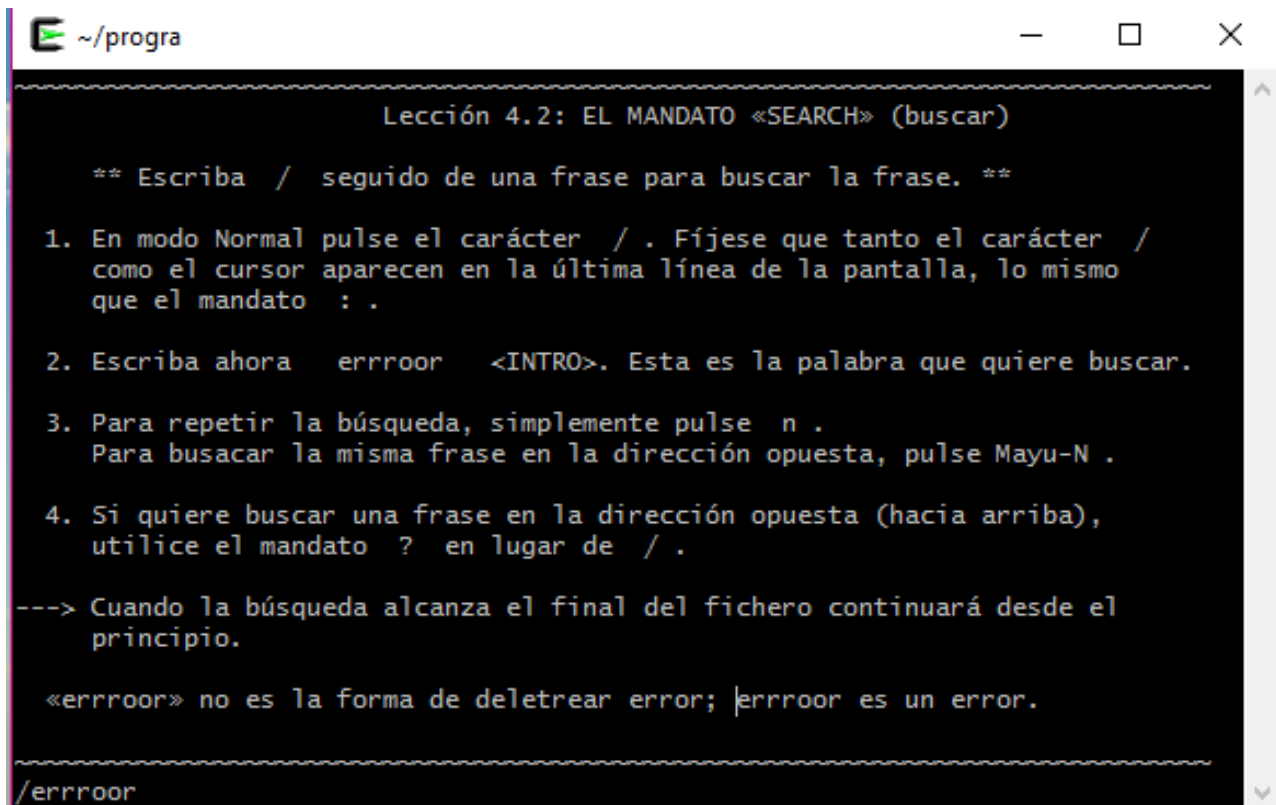
4. Si quiere buscar una frase en la dirección opuesta (hacia arriba),
   utilice el mandato ? en lugar de / .

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

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

search hit BOTTOM, continuing at TOP
```


Utilizando n podremos repetir la búsqueda. Si se requiere que la búsqueda sea por el lado contrario entonces aplicamos el mandato Mayu-N.



```
~/progra

Lección 4.2: EL MANDATO «SEARCH» (buscar)

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

1. En modo Normal pulse el carácter /. Fíjese 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 .

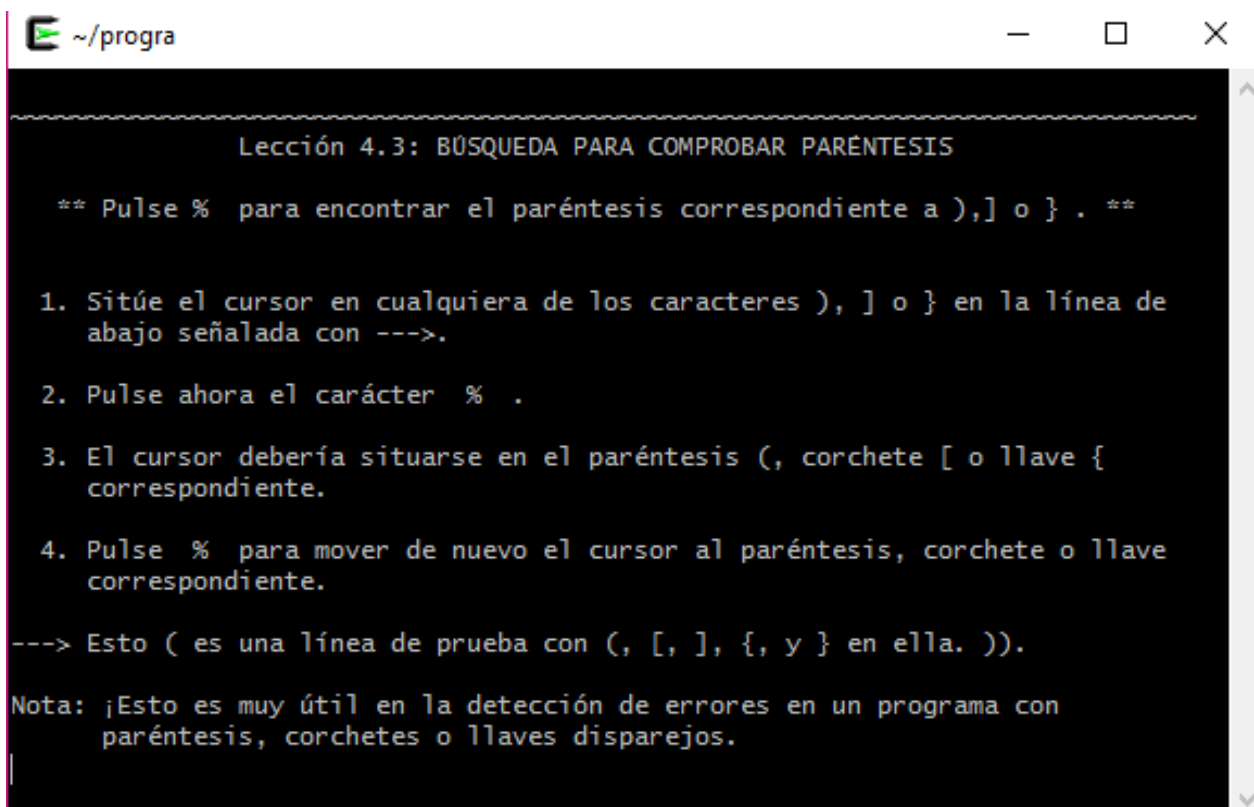
4. Si quiere buscar una frase en la dirección opuesta (hacia arriba),
   utilice el mandato ? en lugar de / .

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

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

/errroor
```

Búsqueda de paréntesis utilizando %.



```
~/progra

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.

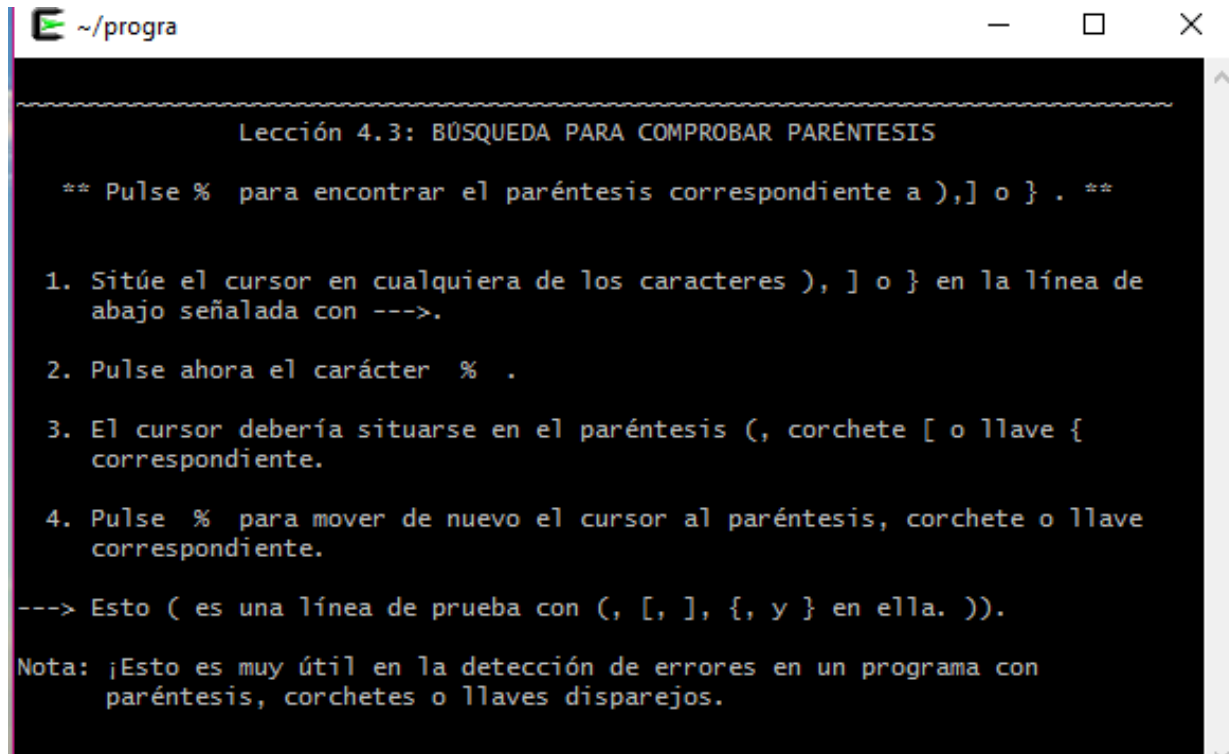
4. Pulse % para mover de nuevo el cursor al paréntesis, corchete o llave
   correspondiente.

---> 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 disparejos.

|
```


Situando el cursor en algunos de los paréntesis de los que deseamos localizar su par, y utilizando el comando % nos llevara al lugar donde se encuentra el par de ese paréntesis. Esto nos hará útil cuando no sepamos cuál de los paréntesis no tiene su respetiva pareja.



```
~/progra

-----
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 % .

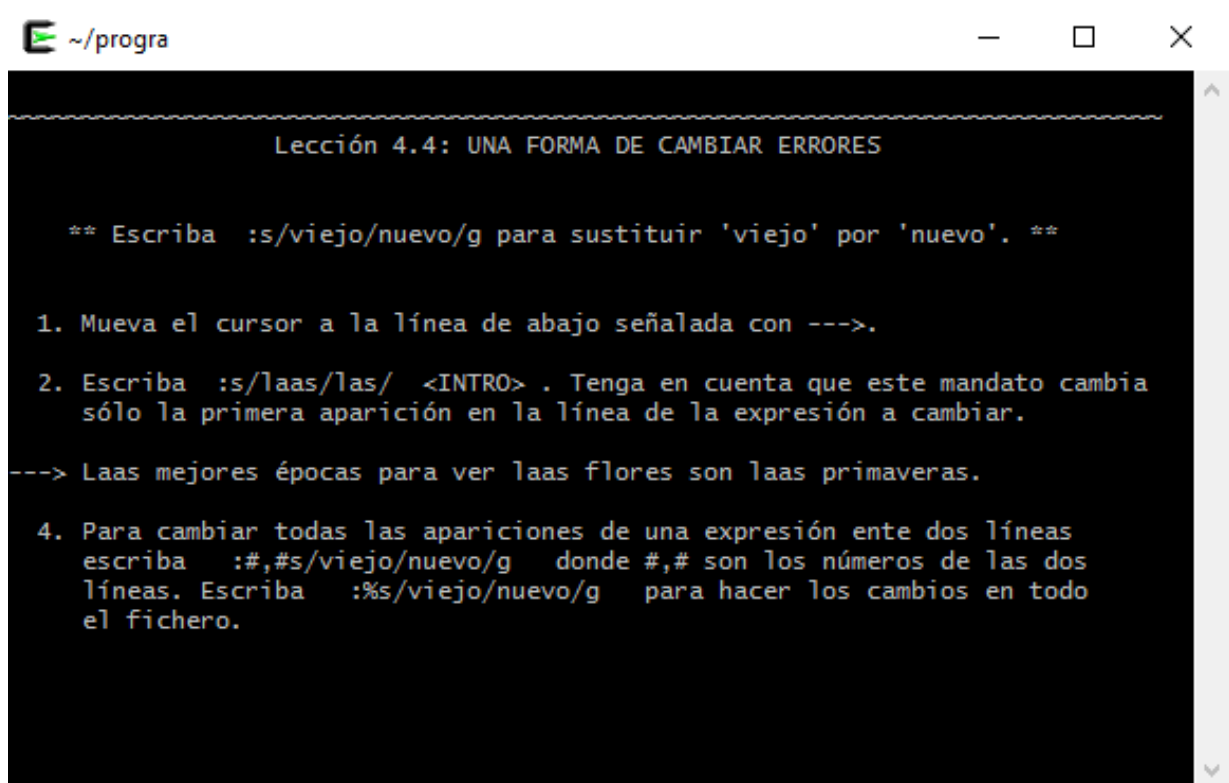
3. El cursor debería situarse en el paréntesis (, corchete [ o llave {
   correspondiente.

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.
```

Sustituyendo una palabra que se encuentra en una línea por otra con el mandato :s/ seguidas por las palabras a cambiar.



```
~/progra

-----
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.
```

Aplicando el mandato.

```
~/progra

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 las 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.

:s/laas/las/
```

Para hacer cambios en todo el fichero aplicaremos :%s/ y las palabras a cambiar.

```
~/progra

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 --->.

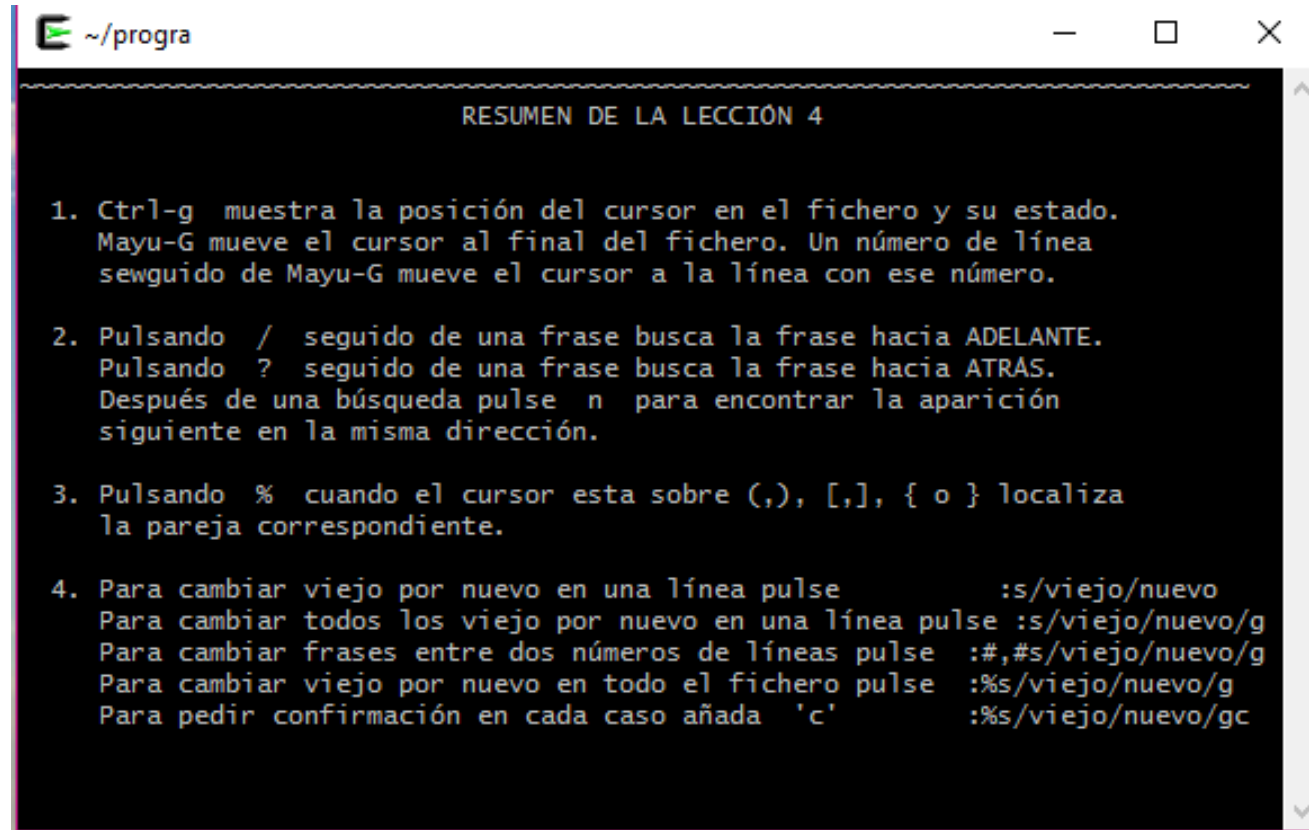
2. Escriba :s/las/las/ <INTRO> . Tenga en cuenta que este mandato cambia
sólo la primera aparición en la línea de la expresión a cambiar.

---> Laas mejores épocas para ver 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.

:%s/laas/las/
```

El resumen nos ayuda a repasar lo aprendido.



```
~/progra

RESUMEN DE LA LECCIÓN 4

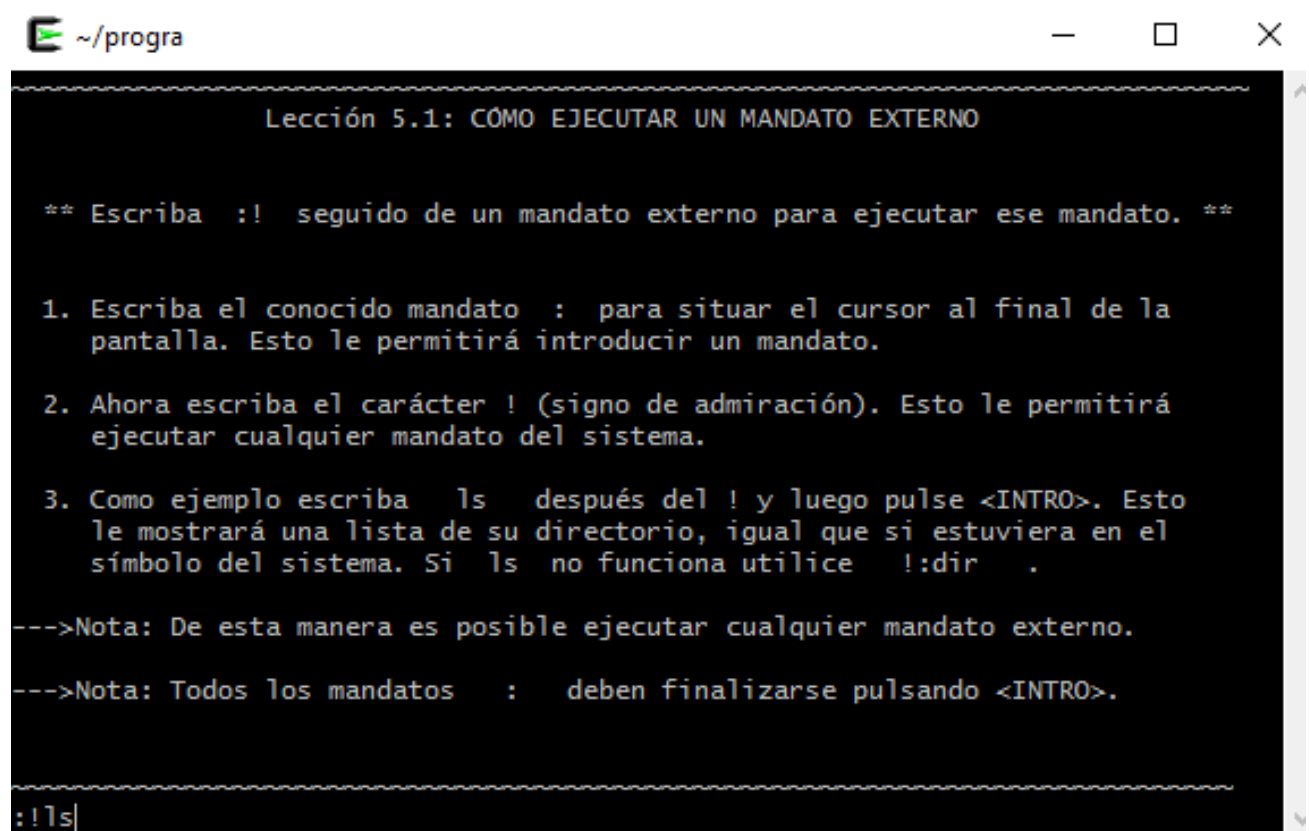
1. Ctrl-g muestra la posición del cursor en el fichero y su estado.
   Mayu-G mueve el cursor al final del fichero. Un número de línea
   seguido de Mayu-G mueve el cursor a la línea con ese número.

2. Pulsando / seguido de una frase busca la frase hacia ADELANTE.
   Pulsando ? seguido de una frase busca la frase hacia ATRÁS.
   Después de una búsqueda pulse n para encontrar la aparición
   siguiente en la misma dirección.

3. Pulsando % cuando el cursor esta sobre (,), [,], { o } localiza
   la pareja correspondiente.

4. Para cambiar viejo por nuevo en una línea pulse           :s/viejo/nuevo
   Para cambiar todos los viejo por nuevo en una línea pulse :s/viejo/nuevo/g
   Para cambiar frases entre dos números de líneas pulse    :#, #s/viejo/nuevo/g
   Para cambiar viejo por nuevo en todo el fichero pulse    :%s/viejo/nuevo/g
   Para pedir confirmación en cada caso añada 'c'           :%s/viejo/nuevo/gc
```

Utilizando el mandato :! Podemos hacer un mandato externo sin tener que salir de nuestro fichero.



```
~/progra

Lección 5.1: COMO EJECUTAR UN MANDATO EXTERNO

** Escriba :! seguido de un mandato externo para ejecutar ese mandato. **

1. Escriba el conocido mandato : para situar el cursor al final de la
   pantalla. Esto le permitirá introducir un mandato.

2. Ahora escriba el carácter ! (signo de admiración). Esto le permitirá
   ejecutar cualquier mandato del sistema.

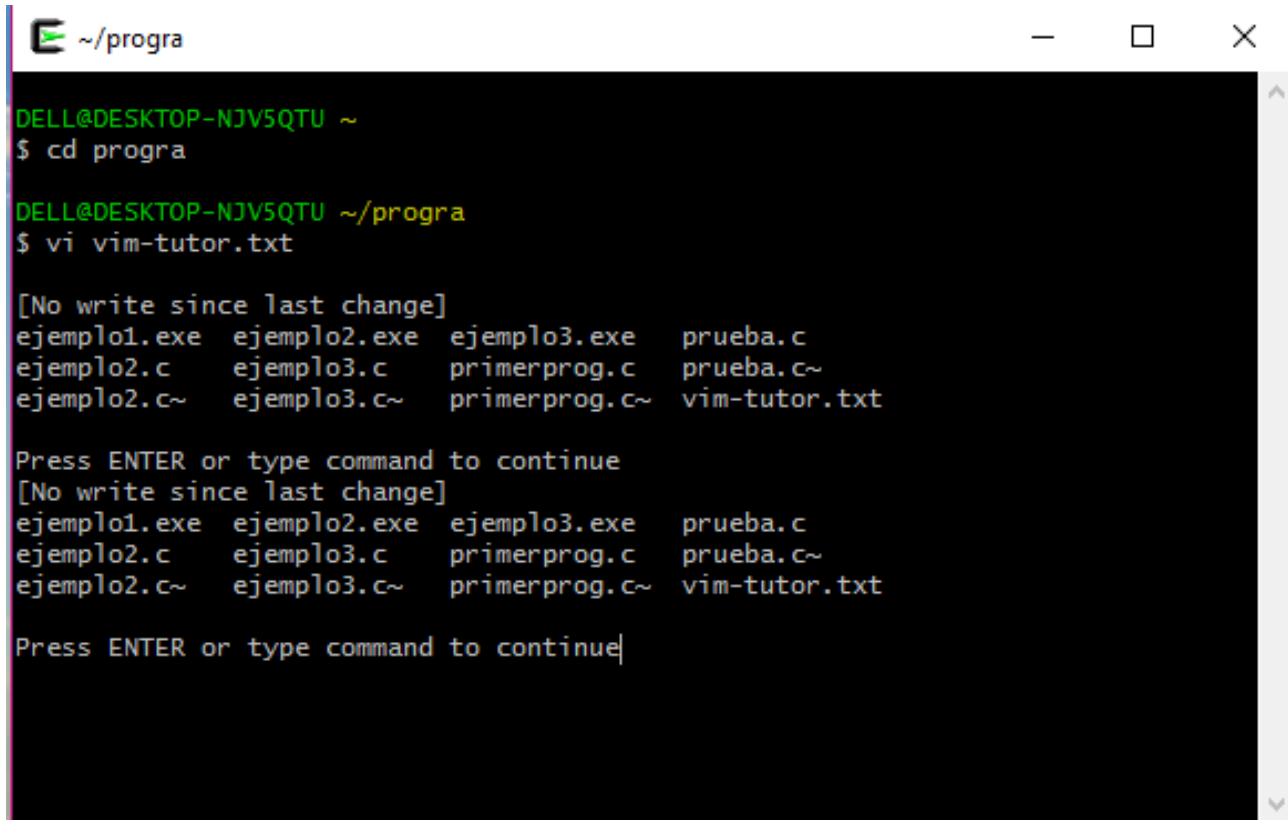
3. Como ejemplo escriba ls después del ! y luego pulse <INTRO>. Esto
   le mostrará una lista de su directorio, igual que si estuviera en el
   símbolo del sistema. Si ls no funciona utilice !:dir .

--->Nota: De esta manera es posible ejecutar cualquier mandato externo.

--->Nota: Todos los mandatos : deben finalizarse pulsando <INTRO>.

:~::~:
:!ls|
```

Aplicando el mandato.



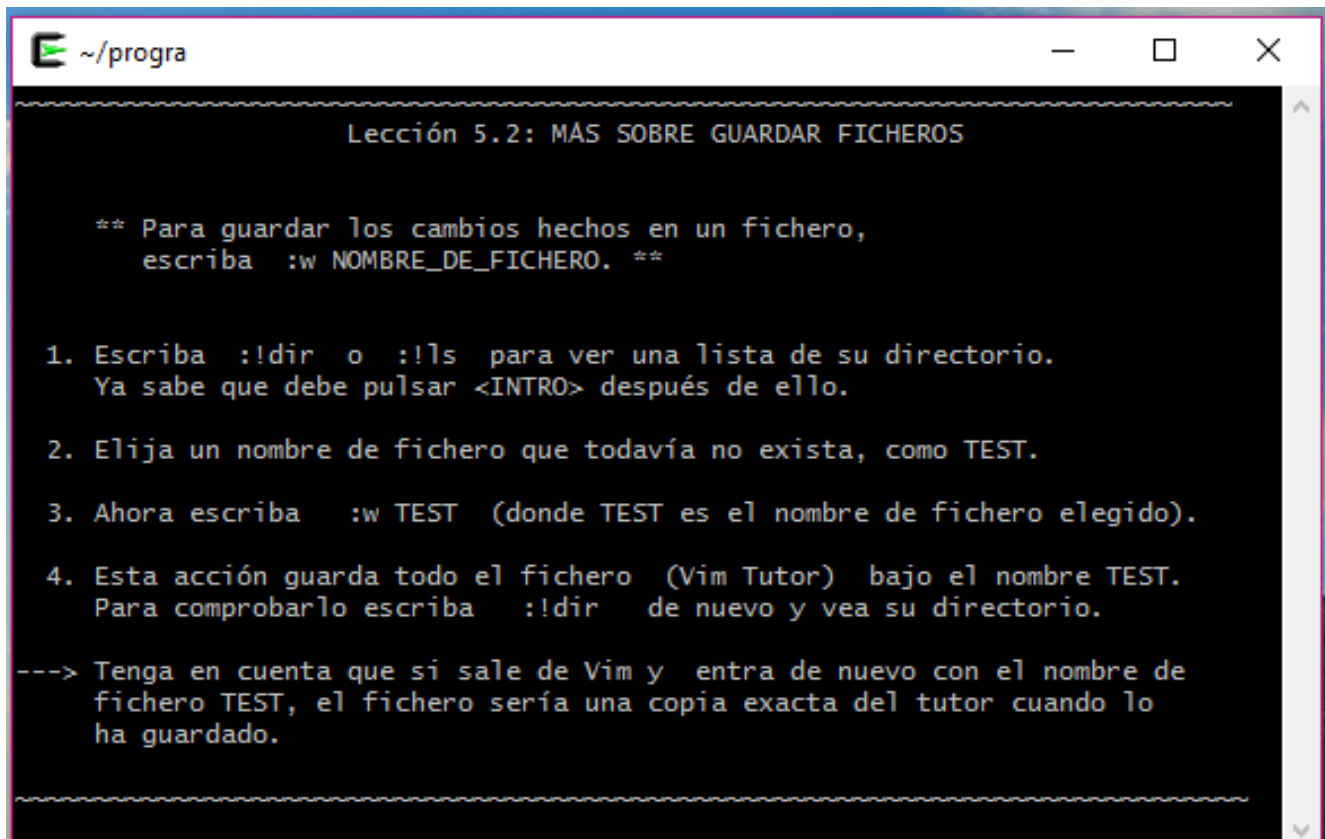
```
~/progra
DELL@DESKTOP-NJV5QTU ~
$ cd progra
DELL@DESKTOP-NJV5QTU ~/progra
$ vi vim-tutor.txt

[No write since last change]
ejemplo1.exe  ejemplo2.exe  ejemplo3.exe  prueba.c
ejemplo2.c    ejemplo3.c    primerprog.c  prueba.c~
ejemplo2.c~   ejemplo3.c~   primerprog.c~ vim-tutor.txt

Press ENTER or type command to continue
[No write since last change]
ejemplo1.exe  ejemplo2.exe  ejemplo3.exe  prueba.c
ejemplo2.c    ejemplo3.c    primerprog.c  prueba.c~
ejemplo2.c~   ejemplo3.c~   primerprog.c~ vim-tutor.txt

Press ENTER or type command to continue|
```

Para guardar un fichero utilizamos el mandato :w seguido del nombre del fichero a guardar.



```
~/progra

=====
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.

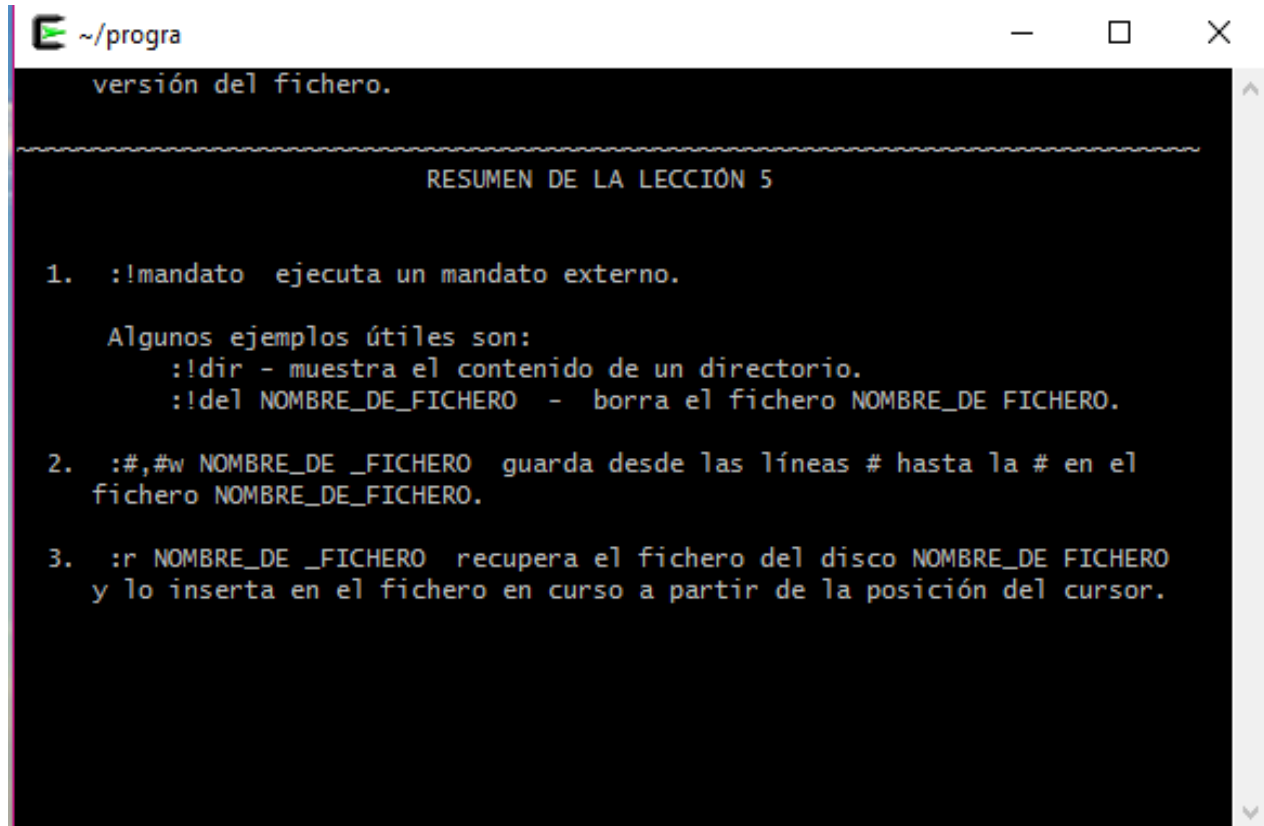
3. Ahora escriba :w TEST (donde TEST es el nombre de fichero elegido).

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 sería una copia exacta del tutor cuando lo
     ha guardado.

=====
```

Resumiendo, como guardar el fichero.



```
~/progra
versión del fichero.

-----
RESUMEN DE LA LECCIÓN 5

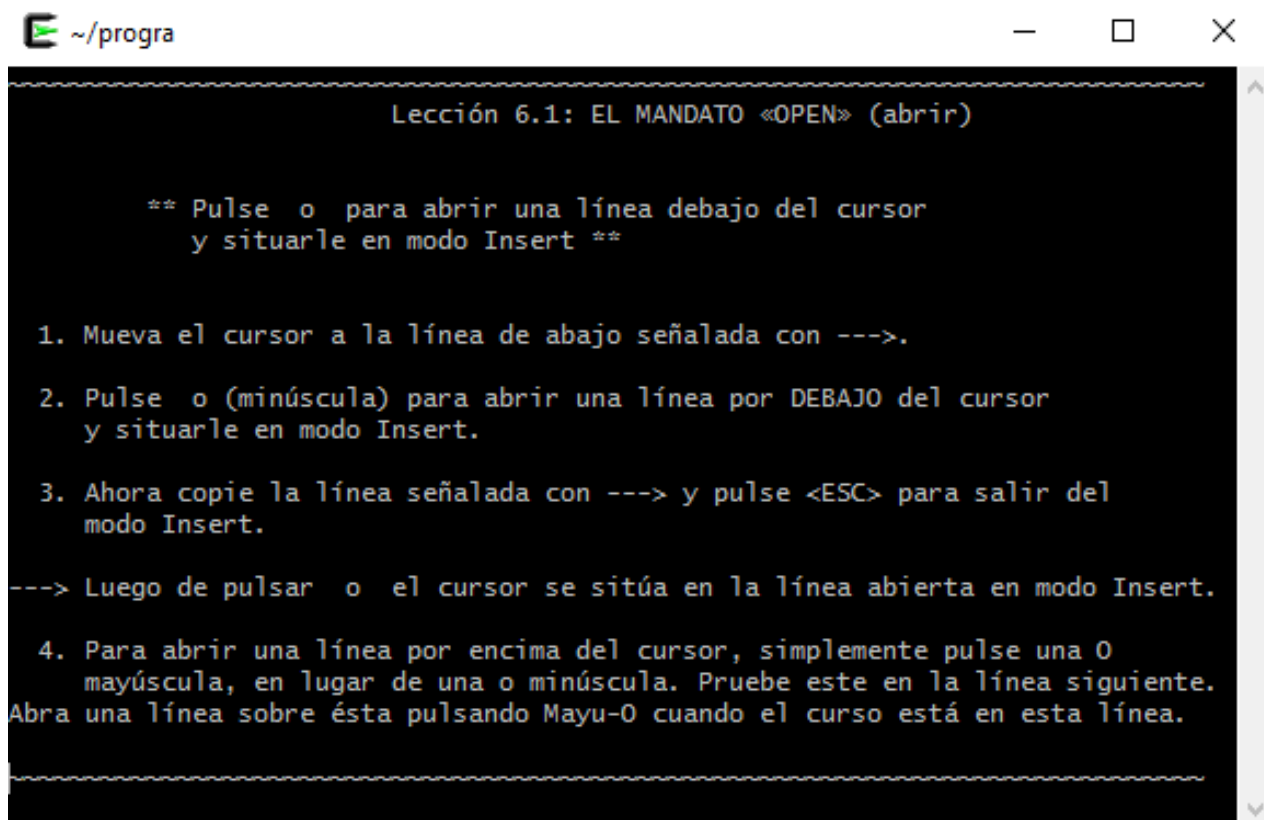
1.  :!mandato  ejecuta un mandato externo.

    Algunos ejemplos útiles son:
        :!dir - muestra el contenido de un directorio.
        :!del NOMBRE_DE_FICHERO - borra el fichero NOMBRE_DE_FICHERO.

2.  :#,#w NOMBRE_DE_FICHERO guarda desde las líneas # hasta la # en el
    fichero NOMBRE_DE_FICHERO.

3.  :r NOMBRE_DE_FICHERO recupera el fichero del disco NOMBRE_DE_FICHERO
    y lo inserta en el fichero en curso a partir de la posición del cursor.
```

El mandato “o” nos abre una línea debajo del cursor donde estamos en modo de insertar.



```
~/progra

-----
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.

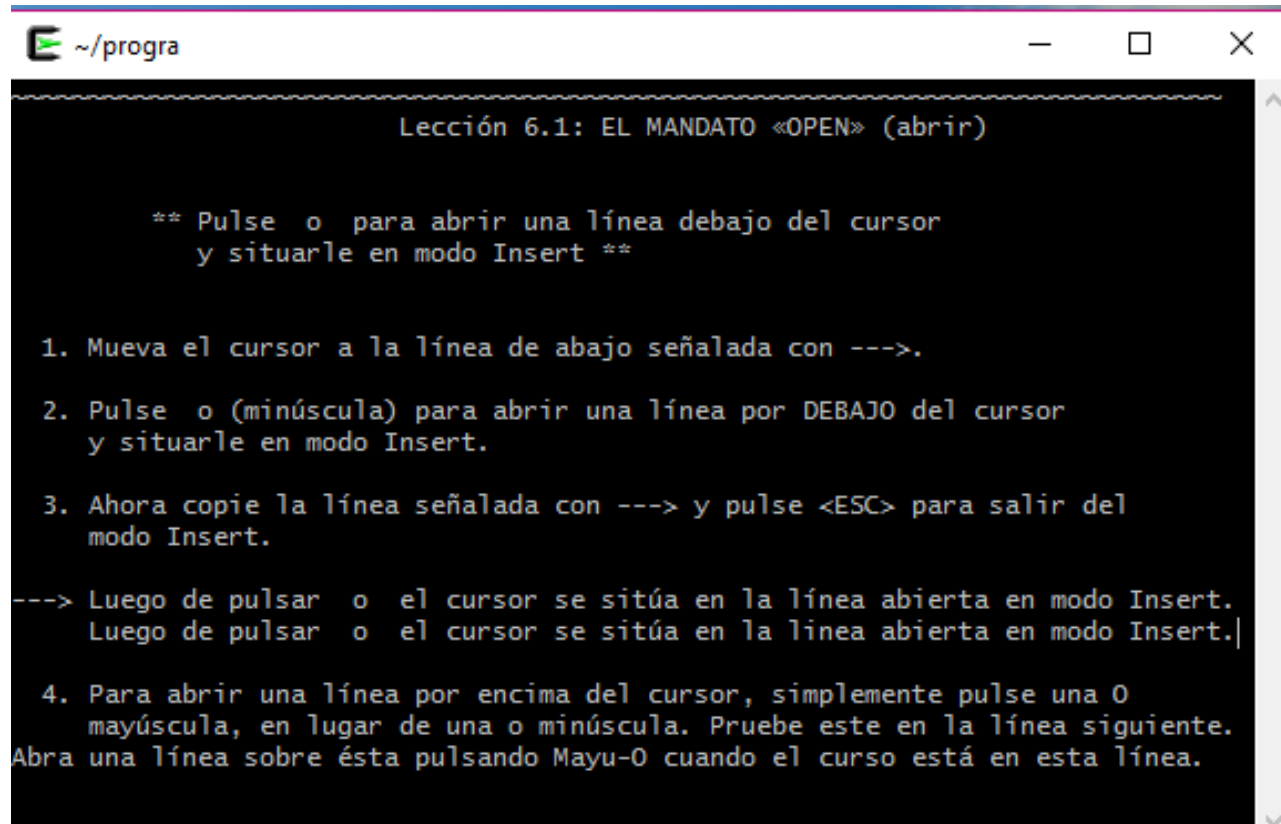
3.  Ahora copie la línea señalada con ---> y pulse <ESC> para salir del
    modo Insert.

---> Luego de pulsar o el cursor se sitúa en la línea abierta en modo Insert.

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.

-----
```

Aplicando el mandato.



```
~/progra
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 --->.

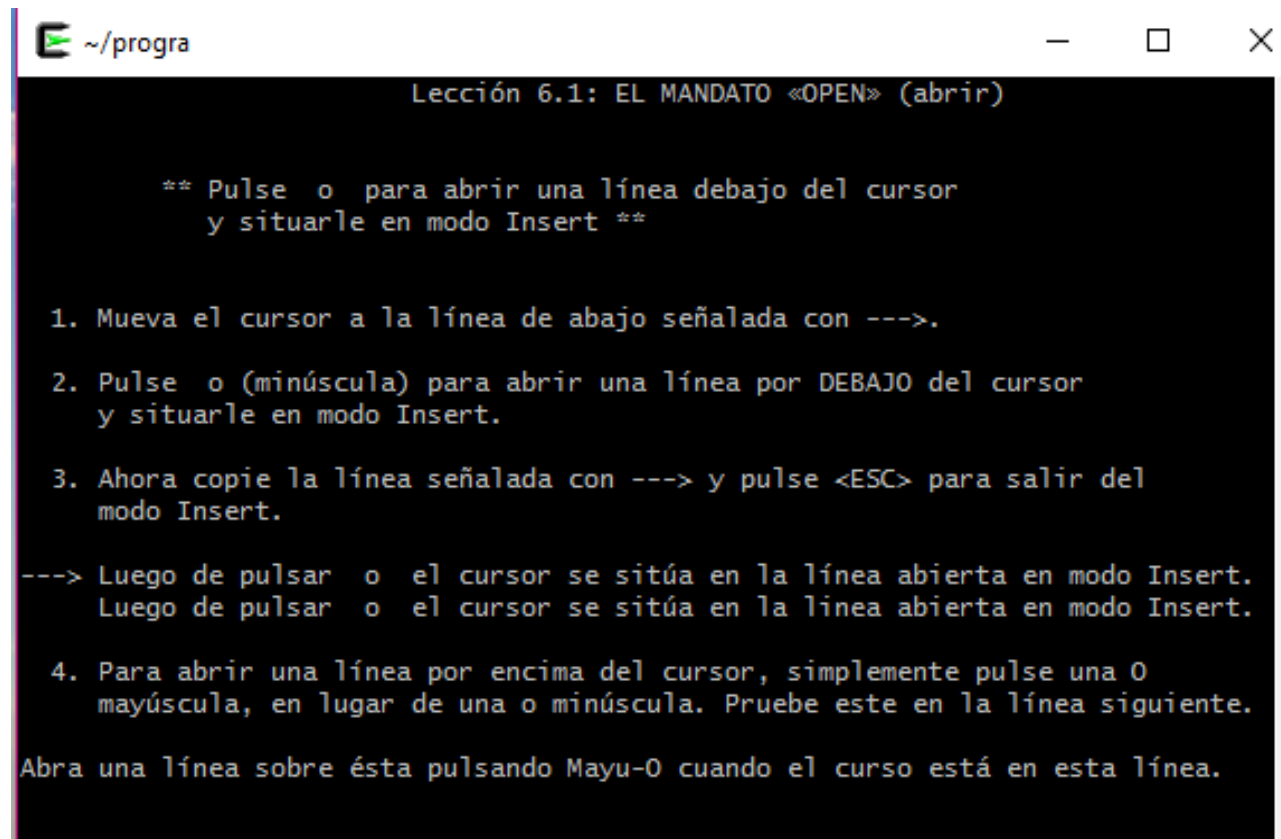
2. Pulse o (minúscula) para abrir una línea por DEBAJO del cursor
y situarle en modo Insert.

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 sitúa 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.
```

Y utilizando una O mayúscula nos abre una línea arriba de donde este el cursor.



```
~/progra
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 --->.

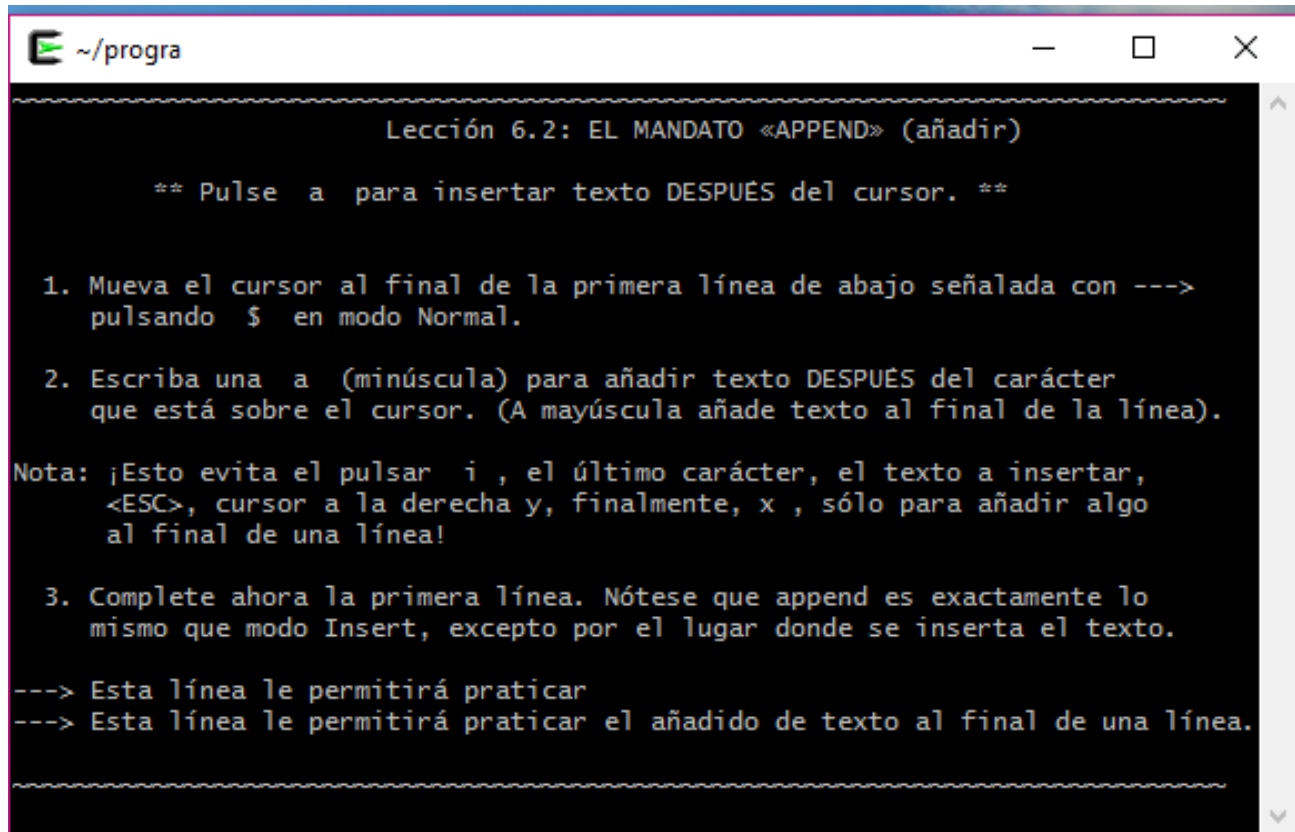
2. Pulse o (minúscula) para abrir una línea por DEBAJO del cursor
y situarle en modo Insert.

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 sitúa 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.
```

El mandato \$a nos permite añadir texto después de donde se situó el cursor.



```
~/progra

Lección 6.2: EL MANDATO «APPEND» (añadir)

** Pulse a para insertar texto DESPUÉS del cursor. **

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

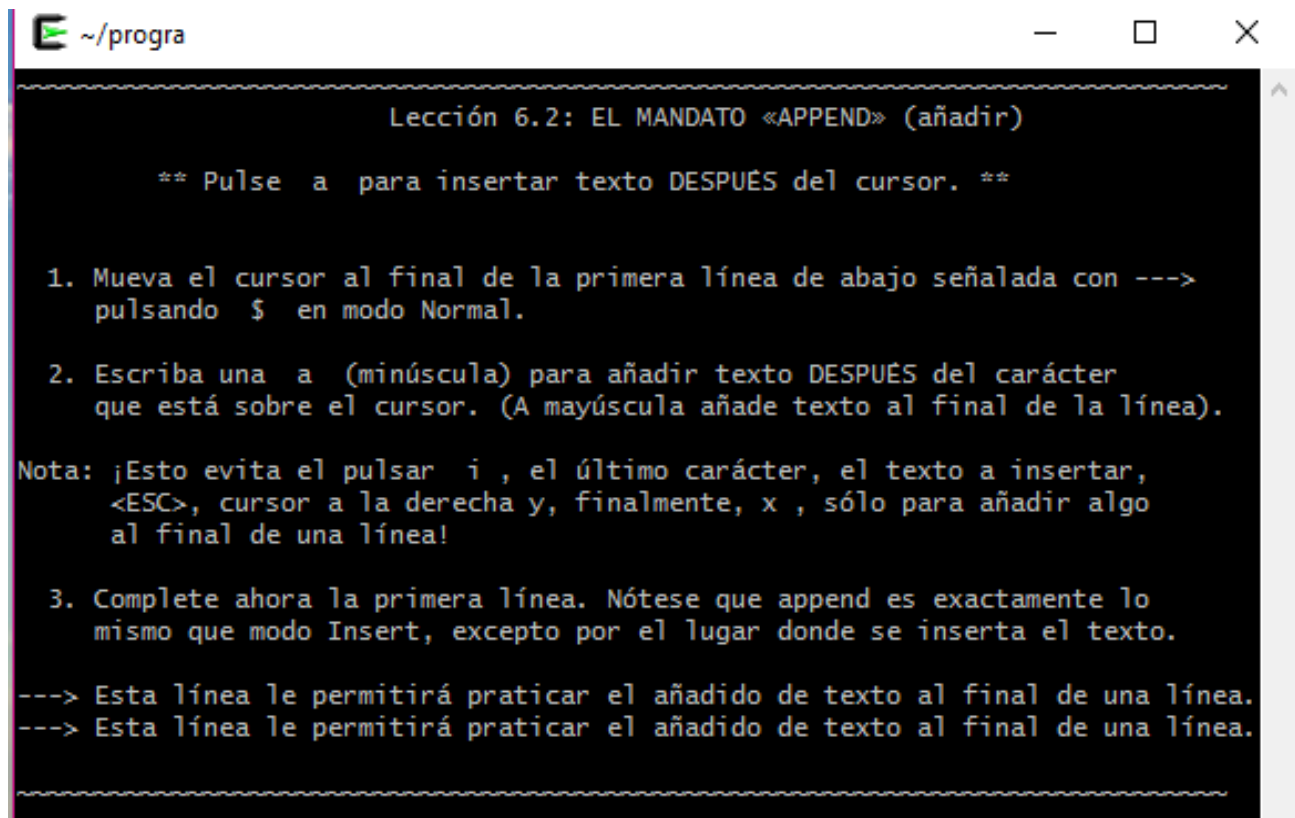
2. Escriba una a (minúscula) para añadir texto DESPUÉS del carácter
   que está sobre el cursor. (A mayúscula añade texto al final de la línea).

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.
```

Aplicando lo aprendido, tenemos que es mas fácil que insert "i".



```
~/progra

Lección 6.2: EL MANDATO «APPEND» (añadir)

** Pulse a para insertar texto DESPUÉS del cursor. **

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

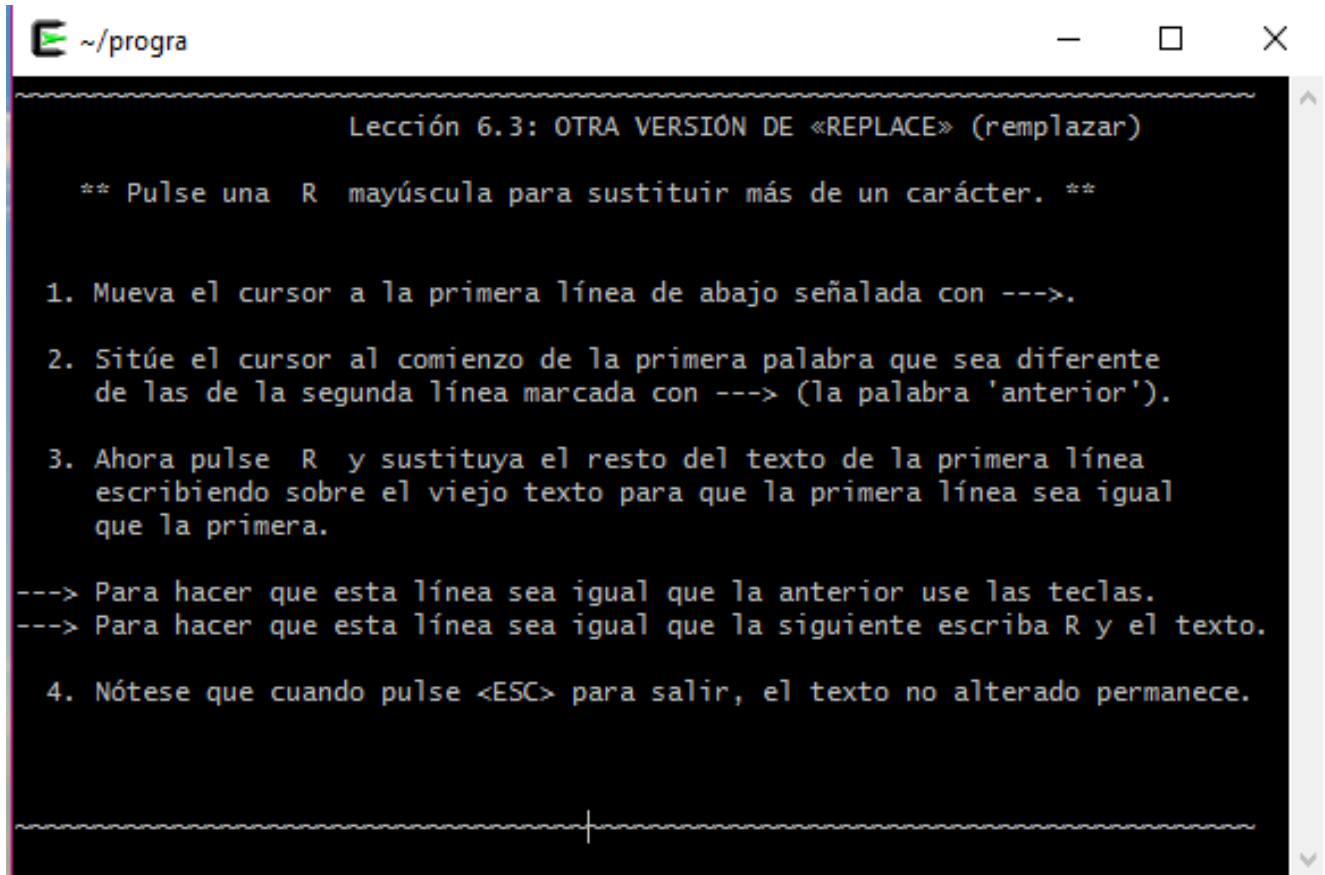
2. Escriba una a (minúscula) para añadir texto DESPUÉS del carácter
   que está sobre el cursor. (A mayúscula añade texto al final de la línea).

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.
```


El mandato R reemplaza más de un carácter.



A terminal window titled "~/progra" with standard window controls. The content is a lesson titled "Lección 6.3: OTRA VERSIÓN DE «REPLACE» (reemplazar)". It explains the use of the uppercase R command to replace multiple characters. The text includes instructions on cursor movement and the use of the R key. At the bottom, there is a dashed line with a vertical cursor positioned in the middle.

```
~/progra

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').

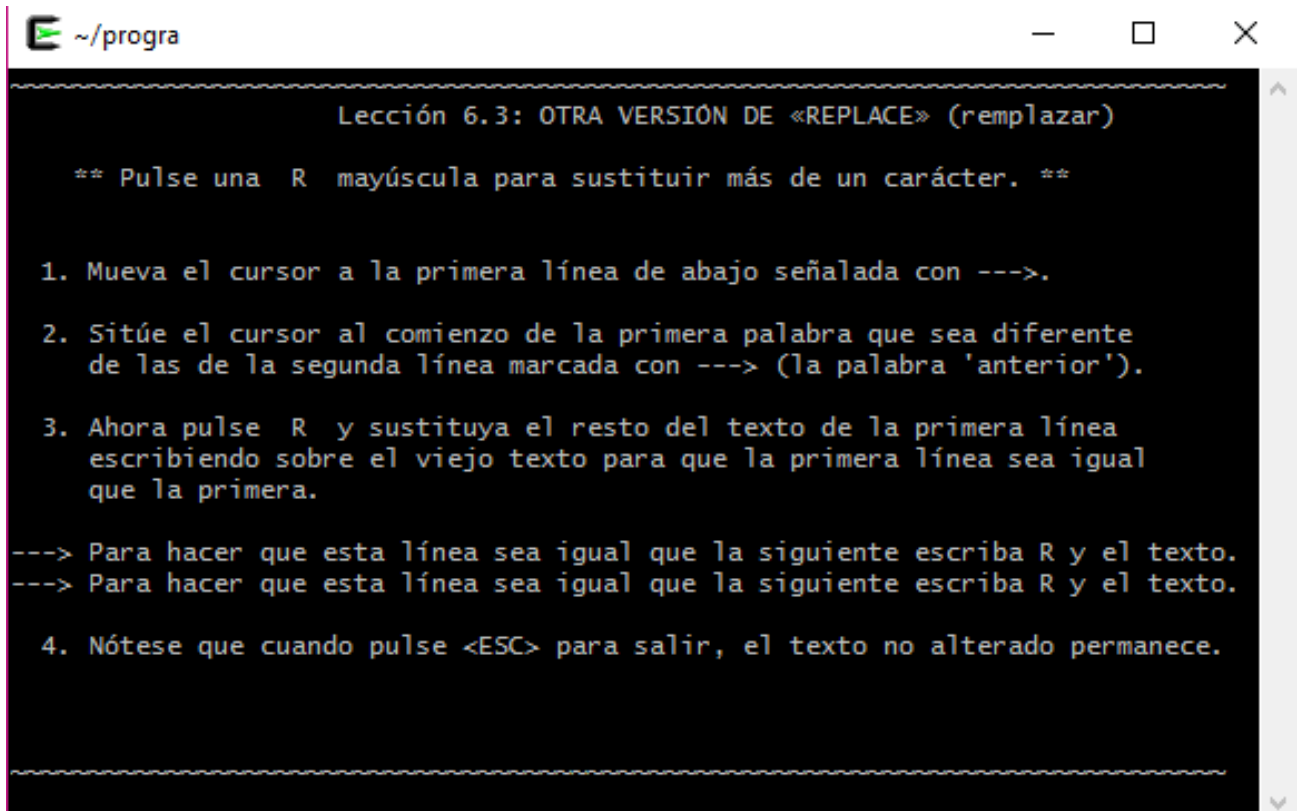
3. Ahora pulse R y sustituya el resto del texto de la primera línea
   escribiendo sobre el viejo texto para que la primera línea sea igual
   que la primera.

---> Para hacer que esta línea sea igual que la anterior use las teclas.
---> 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.

-----|-----
```

Aplicando R tenemos que podemos escribir sobre el texto que ya está, y así reemplazarlo.



The same terminal window as above, but now showing the result of applying the R command. The text is identical to the previous screenshot, but the two lines that were previously marked with ---> are now identical: "Para hacer que esta línea sea igual que la siguiente escriba R y el texto." The vertical cursor is no longer visible at the bottom.

```
~/progra

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').

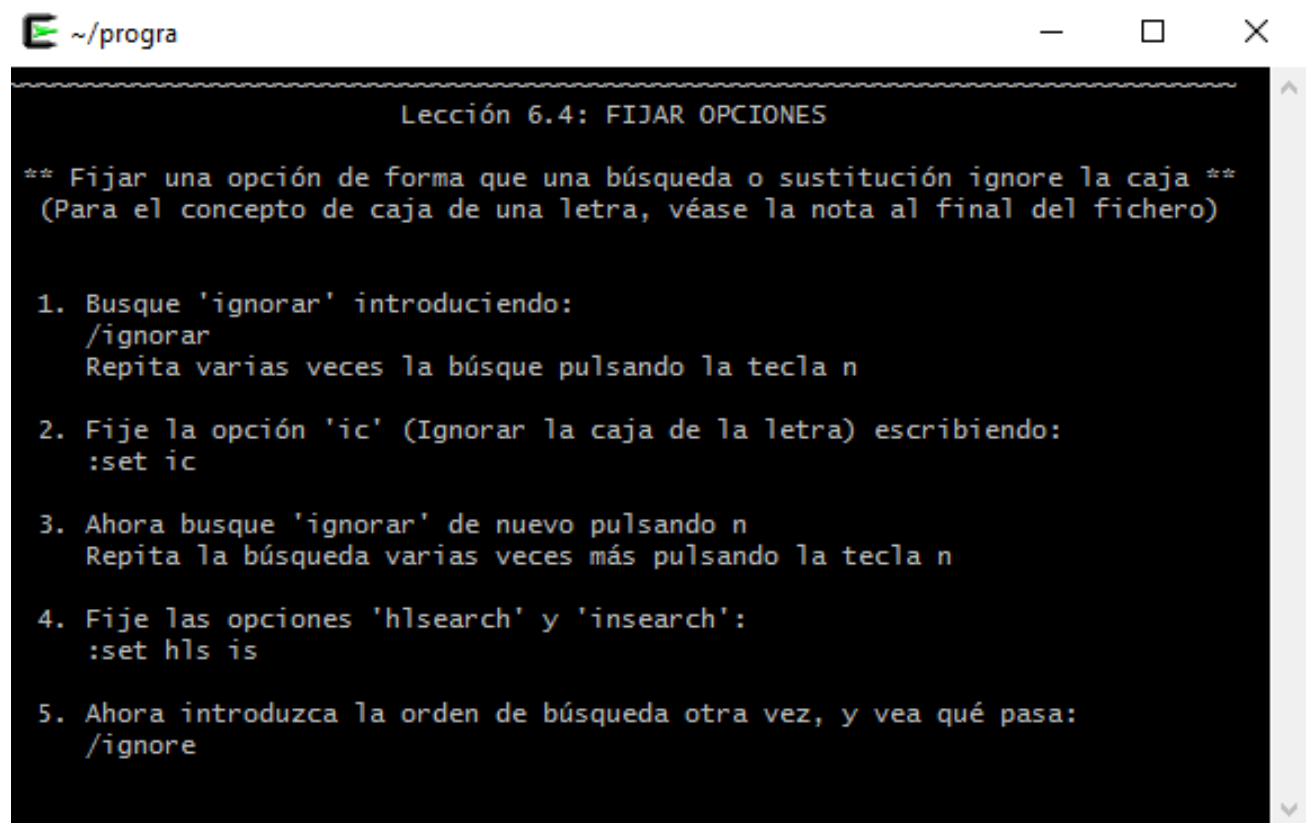
3. Ahora pulse R y sustituya el resto del texto de la primera línea
   escribiendo sobre el viejo texto para que la primera línea sea igual
   que la primera.

---> 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.

-----
```


Mandato para fijar opciones de búsqueda. (No encontré exactamente que es lo que tenía que suceder.



```
Lección 6.4: FIJAR OPCIONES

** 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

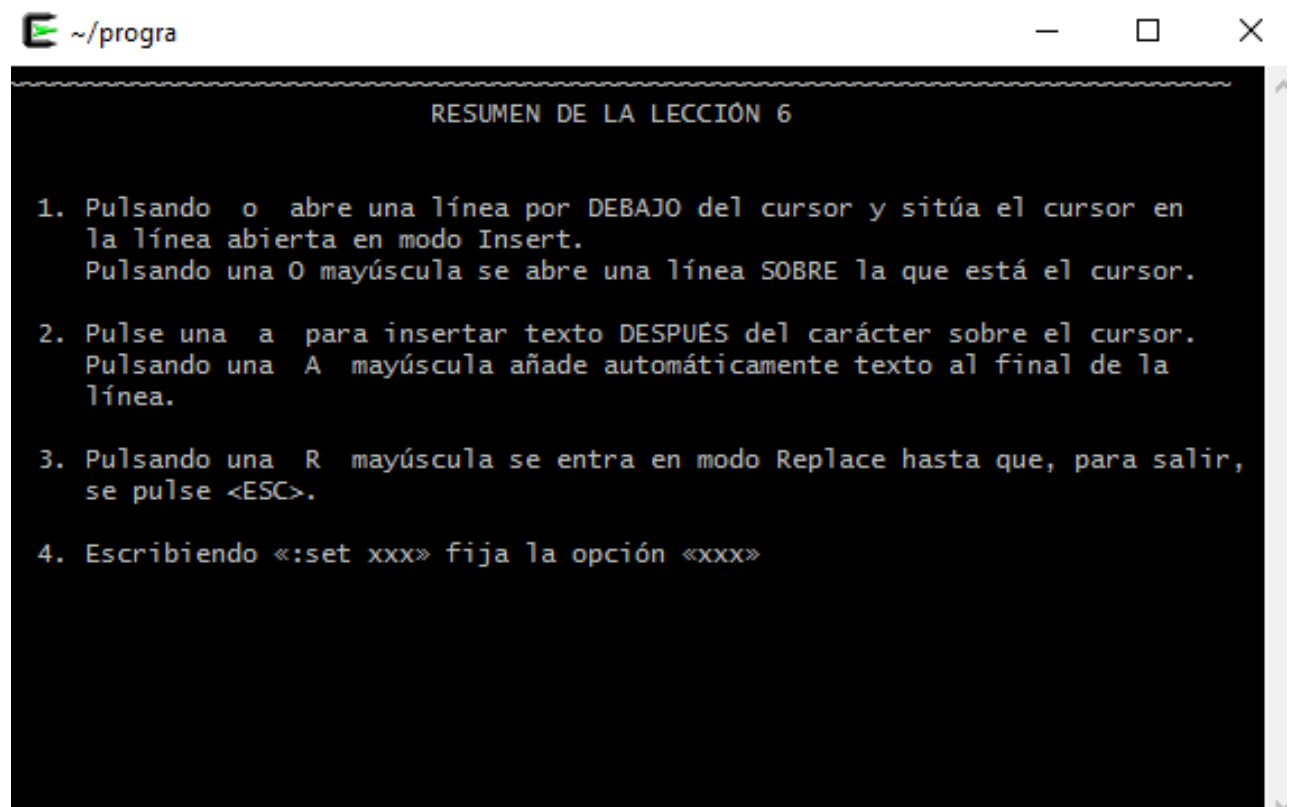
2. Fije la opción 'ic' (Ignorar la caja de la letra) escribiendo:
   :set ic

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
```

Resumiendo, la lección.



```
RESUMEN DE LA LECCIÓN 6

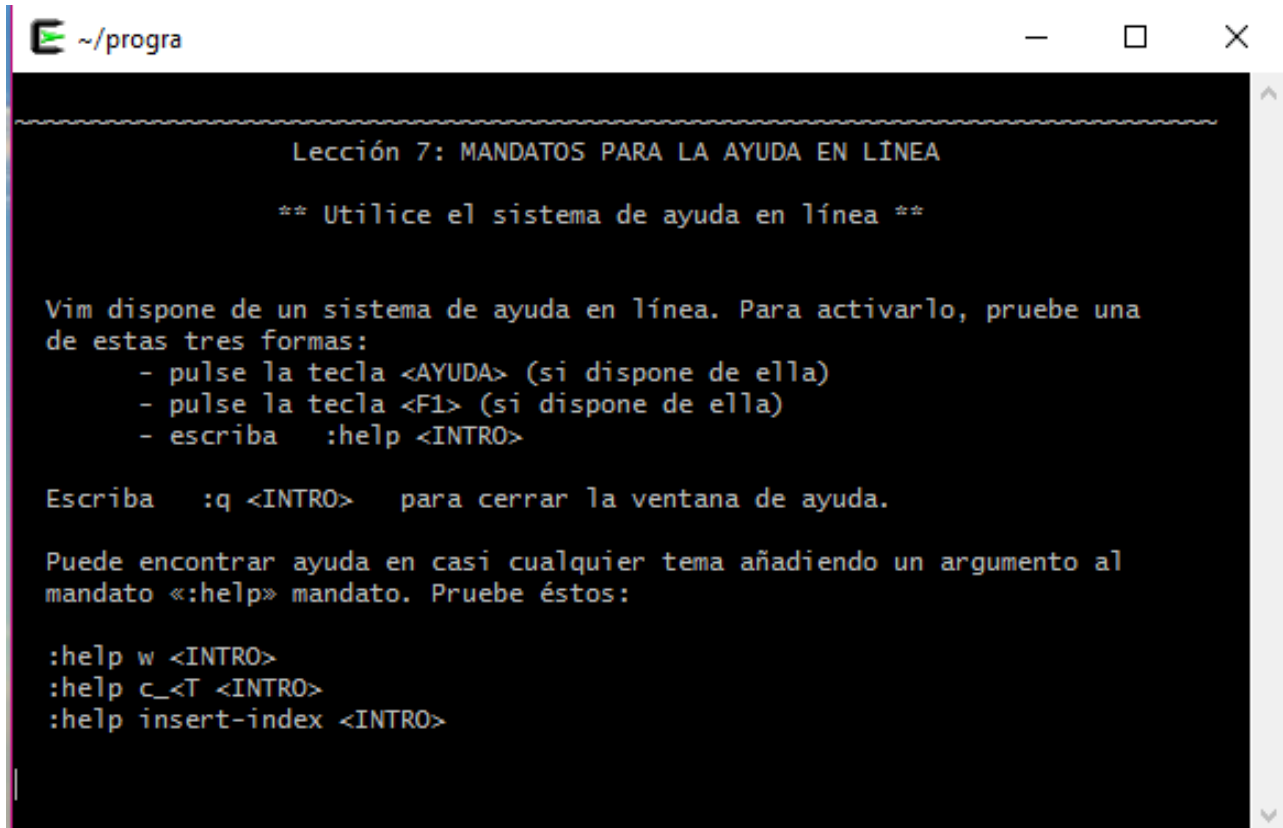
1. Pulsando o abre una línea por DEBAJO del cursor y sitúa el cursor en
   la línea abierta en modo Insert.
   Pulsando una O mayúscula se abre una línea SOBRE la que está el cursor.

2. Pulse una a para insertar texto DESPUÉS del carácter sobre el cursor.
   Pulsando una A mayúscula añade automáticamente texto al final de la
   línea.

3. Pulsando una R mayúscula se entra en modo Replace hasta que, para salir,
   se pulse <ESC>.

4. Escribiendo «:set xxx» fija la opción «xxx»
```

Mandato para ayuda en línea.



The screenshot shows a terminal window with a title bar indicating the path ~/progra. The terminal content is the Vim help system's introductory text, which explains how to use the online help system. It lists three ways to activate help: pressing a key (AYUDA or F1), or typing :help followed by an introduction. It also shows how to close the help window with :q and how to search for help on specific topics like 'w', 'c_T', or 'insert-index'.

```
~/progra

-----
Lección 7: MANDATOS PARA LA AYUDA EN LÍNEA

** Utilice el sistema de ayuda en línea **

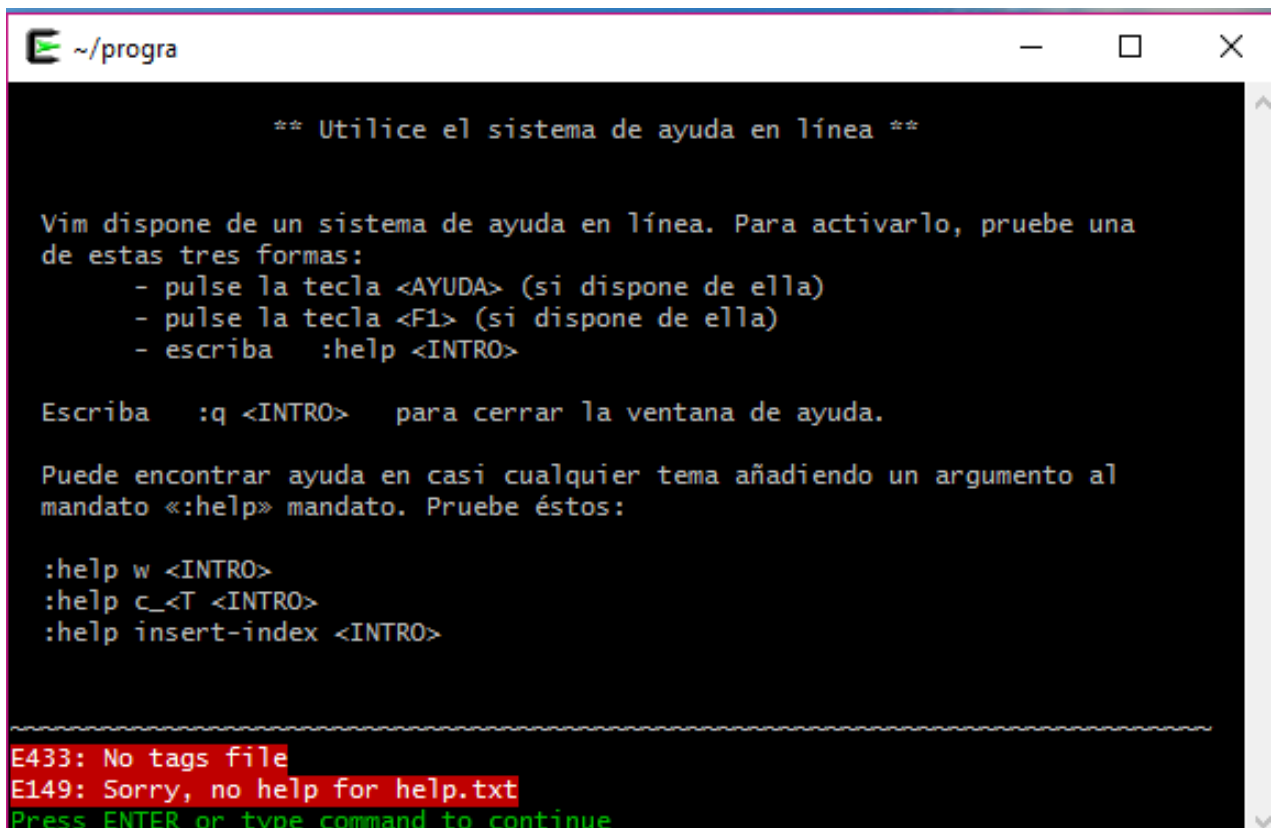
Vim dispone de un sistema de ayuda en línea. Para activarlo, pruebe una
de estas tres formas:
  - pulse la tecla <AYUDA> (si dispone de ella)
  - pulse la tecla <F1> (si dispone de ella)
  - escriba   :help <INTRO>

Escriba   :q <INTRO>   para cerrar la ventana de ayuda.

Puede encontrar ayuda en casi cualquier tema añadiendo un argumento al
mandato «:help» mandato. Pruebe éstos:

:help w <INTRO>
:help c_T <INTRO>
:help insert-index <INTRO>
```

En este caso no me funciono ninguna de las opciones.



This screenshot shows the same terminal window as the previous one, but now it displays error messages from the Vim help system. The messages indicate that the help system cannot find the tags file (E433) and that there is no help available for the help.txt file (E149). The terminal prompts the user to press ENTER or type a command to continue.

```
~/progra

** Utilice el sistema de ayuda en línea **

Vim dispone de un sistema de ayuda en línea. Para activarlo, pruebe una
de estas tres formas:
  - pulse la tecla <AYUDA> (si dispone de ella)
  - pulse la tecla <F1> (si dispone de ella)
  - escriba   :help <INTRO>

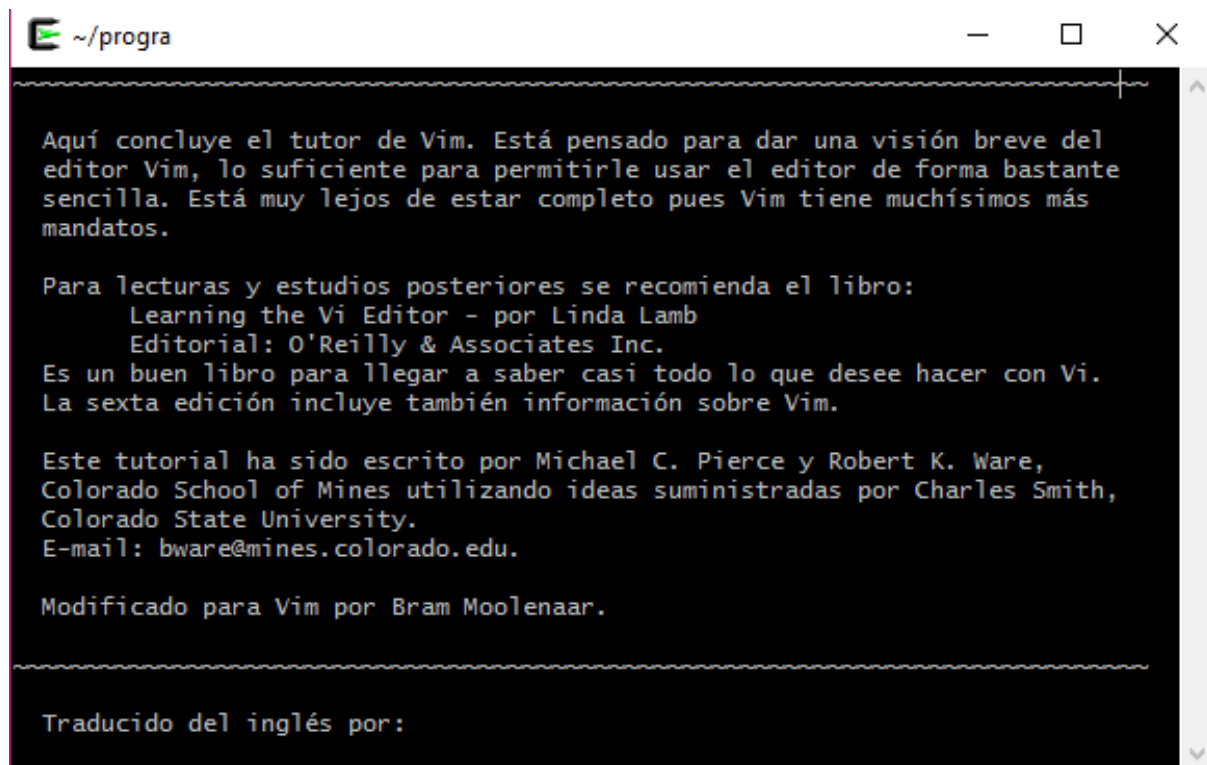
Escriba   :q <INTRO>   para cerrar la ventana de ayuda.

Puede encontrar ayuda en casi cualquier tema añadiendo un argumento al
mandato «:help» mandato. Pruebe éstos:

:help w <INTRO>
:help c_T <INTRO>
:help insert-index <INTRO>

-----
E433: No tags file
E149: Sorry, no help for help.txt
Press ENTER or type command to continue
```

Y finalmente esto es todo acerca de cómo usar lo mejor posible Vim.



A screenshot of a terminal window with a black background and light green text. The window title is "~/progra". The text inside the terminal is as follows:

```
Aquí concluye el tutor de Vim. Está pensado para dar una visión breve del
editor Vim, lo suficiente para permitirle usar el editor de forma bastante
sencilla. Está muy lejos de estar completo pues Vim tiene muchísimos más
mandatos.

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.

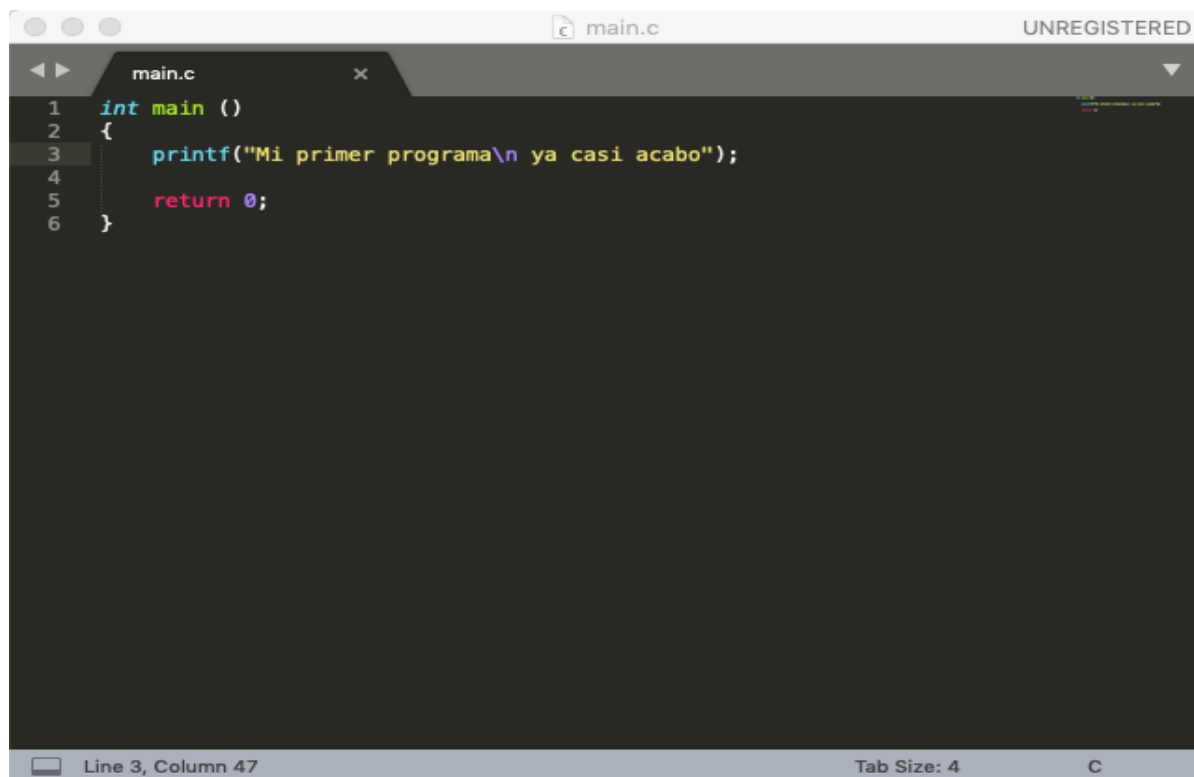
Modificado para Vim por Bram Moolenaar.
```

Below this text is a horizontal dashed line, followed by the text:

```
Traducido del inglés por:
```

Actividad 3.

Haciendo un programa sencillo en lenguaje C con el editor de Sublime Text (Hecho en clase de practica).



A screenshot of the Sublime Text editor interface. The window title is "main.c" and it says "UNREGISTERED" in the top right corner. The code in the editor is:

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

The status bar at the bottom shows "Line 3, Column 47", "Tab Size: 4", and "C".

Y su compilación.

```

Documentos — -bash — 80x24
'int (const char *, ...)' [-Wimplicit-function-declaration]
  printf("Mi primer programa\n ya casi acabo");
  ^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
[Laos03:Documents fp03alu26$ ./main
Mi primer programa
 ya casi acaboLaos03:Documents fp03alu26$ cd ..
[Laos03:~ fp03alu26$ cd Documents
[Laos03:Documents fp03alu26$ ls
main    main.c  main_
[Laos03:Documents fp03alu26$ 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 ya casi acabo\n");
  ^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
[Laos03:Documents fp03alu26$ ./main
Mi primer programa
 ya casi acabo
Laos03:Documents fp03alu26$
```

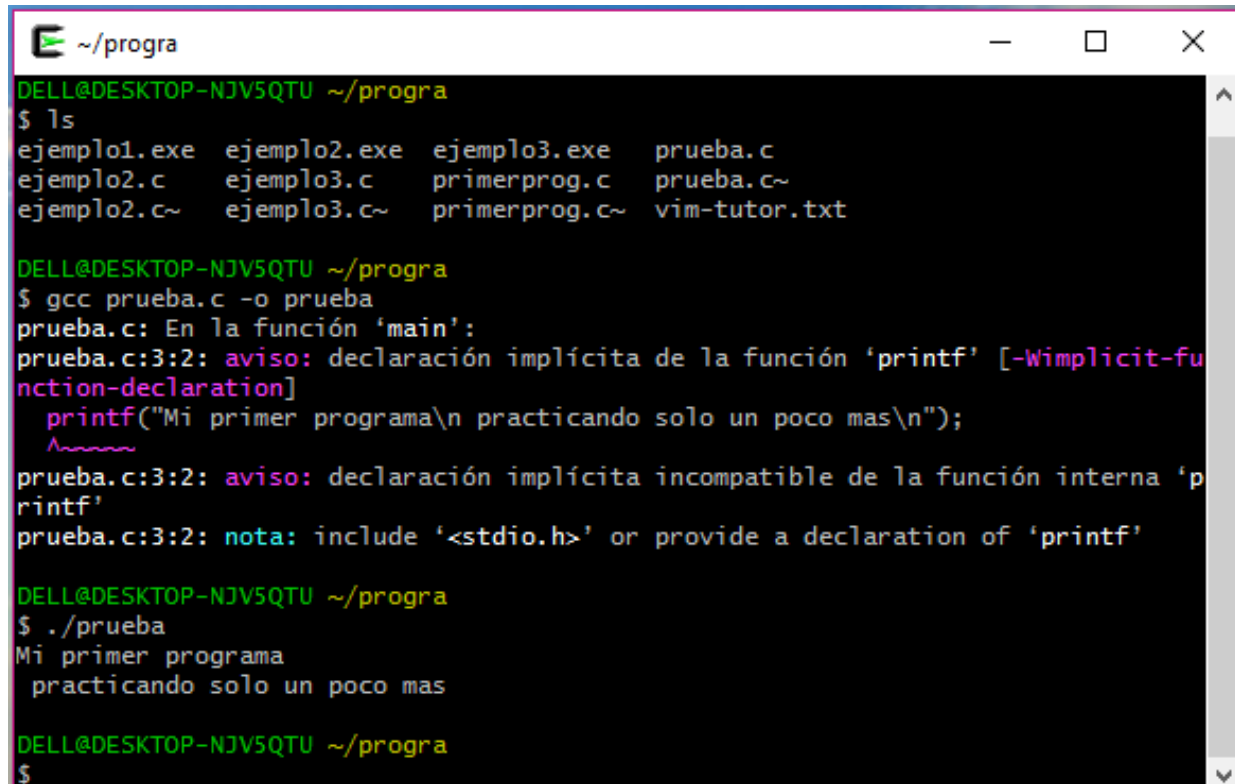
El mismo programa, pero hecho en mi casa en el editor de Vim.

A screenshot of a Windows Notepad++ application window titled "prueba.c~ (D:\fundamentos de pro...cas hechas\programas en c) - GVIM". The menu bar includes Archivo, Editar, Herramientas, Sintaxis, Buffers, Ventana, and Ayuda. The toolbar contains various icons for file operations and editing. The code editor shows a C program with syntax highlighting:

```
int main()
{
    printf("Mi primer programa\n practicando solo un poco mas\n");
    return 0;
}
```

 Blue squiggly lines are visible under the opening curly brace of the main function. The status bar at the bottom displays "<Practic...cas hechas\programas en c\prueba.c~" 6L, 98C" on the left, "1,1" in the center, and "Todo" on the right.

Y su compilación.

A terminal window titled "~/progra" with standard window controls. It shows the execution of a C program. First, a directory listing is shown. Then, the program is compiled with gcc, resulting in several warnings about implicit function declarations for 'printf'. Finally, the program is executed, displaying the output: "Mi primer programa" and "practicando solo un poco mas".

```
DELL@DESKTOP-NJV5QTU ~/progra
$ ls
ejemplo1.exe  ejemplo2.exe  ejemplo3.exe  prueba.c
ejemplo2.c    ejemplo3.c    primerprog.c  prueba.c~
ejemplo2.c~   ejemplo3.c~   primerprog.c~ vim-tutor.txt

DELL@DESKTOP-NJV5QTU ~/progra
$ gcc prueba.c -o prueba
prueba.c: En la función 'main':
prueba.c:3:2: aviso: declaración implícita de la función 'printf' [-Wimplicit-fu
nction-declaration]
    printf("Mi primer programa\n practicando solo un poco mas\n");
    ~~~~~
prueba.c:3:2: aviso: declaración implícita incompatible de la función interna 'p
rintf'
prueba.c:3:2: nota: include '<stdio.h>' or provide a declaration of 'printf'

DELL@DESKTOP-NJV5QTU ~/progra
$ ./prueba
Mi primer programa
practicando solo un poco mas

DELL@DESKTOP-NJV5QTU ~/progra
$
```

Conclusión:

El tutorial de Vim es bastante largo, pero vale la pena hacer cada uno de los pasos que muestra ya que de esta manera hace más sencillo el manejo de esta herramienta, lo cual es importante para nuestras practica de aquí en adelante, ya que estaremos haciendo programas que necesariamente utilizan estos tipos de editores. La investigación que se realizó en la primera actividad nos ayuda a saber y conocer los diferentes tipos de archivos en texto plano que podemos ocupar más adelante para nuestros trabajos. Y el hacer un programa sencillo en Lenguaje C nos introduce un poco el panorama de lo que estaremos trabajando.