



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

### Laboratorios de computación Salas A y B

---

*Profesor:* Alejandro Pimentel

*Asignatura:* Laboratorio de protraction

*Grupo:* 135

*No de Práctica(s):* Práctica 6

Areli González Segura

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:* 23

*No. de Lista o Brigada:* 5319 no. Lista: 19

*Semestre:* Primer semestre

*Fecha de entrega:* 30/septiembre/2019

*Observaciones:* Desarrollo correcto

CALIFICACIÓN: **10**

# Entorno de C

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

Un archivo de texto simple, texto sencillo o texto sin formato (también llamado texto llano o texto simple; en inglés «plain text»), es un archivo informático que contiene únicamente texto formado solo por caracteres que son legibles por humanos, careciendo de cualquier tipo de formato tipográfico.<sup>1</sup>

Estos archivos están compuestos de bytes que representan caracteres ordinarios como letras, números y signos de puntuación (incluyendo espacios en blanco), también incluye algunos pocos caracteres de control como tabulaciones, saltos de línea y retornos de carro. Estos caracteres se pueden codificar de distintos modos. El sistema de codificación ASCII viene a ser la base primordial y no necesita de un identificador explícito en la comunicación digital. Para poder representar diferentes sistemas de codificación de caracteres como UTF-1, UTF-7, UTF-8, UTF-16, UTF-32, UTF-EBCDIC, SCSU, BOCU-1 y GB 18030 es necesario que haya al principio de cada fichero de texto una marca de orden de bytes.

Los archivos de texto llano carecen de información destinada a generar formatos (negritas, subrayado, cursivas, tamaño, etc.)<sup>2</sup> y tipos de letra (por ejemplo, Arial, Times, Courier, etc.).<sup>2</sup> Esta simplicidad permite que una gran variedad de programas pueda leer y editar ese contenido. Las aplicaciones destinadas a la escritura y modificación de archivos de texto se llaman editores de texto. Cada texto tiene un peso diferente porque tiene una codificación distinta

### Markdown

Markdown es un lenguaje de marcado ligero creado por John Gruber que trata de conseguir la máxima legibilidad y facilidad de publicación tanto en su forma de entrada como de salida, inspirándose en muchas convenciones existentes para marcar mensajes de correo electrónico usando texto plano. Se distribuye bajo licencia BSD y se distribuye como plugin (o al menos está disponible) en diferentes sistemas de gestión de contenidos (CMS). Markdown convierte el texto marcado en documentos XHTML utilizando html2text creado por Aaron Swartz. Markdown fue implementado originariamente en Perl por Gruber, pero desde entonces ha sido traducido a multitud de lenguajes de programación,

incluyendo PHP, Python, Ruby, Java y Common Lisp

## **Html**

Un documento HTML es un archivo de texto que tienen la extensión .html o .htm, este tipo de archivo se puede abrir con un editor de textos para editar el código o con un navegador de Internet para visualizar la página resultado del código HTML escrito.

Para desarrollar una página web en HTML es necesario crear un documento HTML.

Básicamente un documento HTML es un archivo de texto que tienen la extensión .html o .htm, en este documento se escriben todo el texto y las etiquetas HTML necesarias para la creación de una página, al texto escrito en el documento HTML se le llama código HTML. Un documento HTML se puede generar con cualquier editor de textos simple como el bloc de notas de Windows o Gedit de Linux.

## **LaTeX**

Es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.

LaTeX está formado por un gran conjunto de macros de TeX, escrito por Leslie Lamport en 1984, con la intención de facilitar el uso del lenguaje de composición

tipográfica, creado por Donald Knuth. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados en LaTeX, se considera adecuada a las necesidades de una editorial científica de primera línea, muchas de las cuales ya lo emplean.

LaTeX es software libre bajo licencia LPPL.

## **Csv**

La abreviatura CSV se refiere a un tipo de datos de texto en el procesamiento de datos con el que se pueden registrar, almacenar y procesar grandes cantidades de datos estructurados. El término CSV significa "valores separados por comas" (comma-separated values). Los archivos CSV se utilizan con frecuencia para los catálogos de productos de las tiendas online. El archivo termina en .csv.

## Editores

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

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

etc.

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

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

etc.

### Nano

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

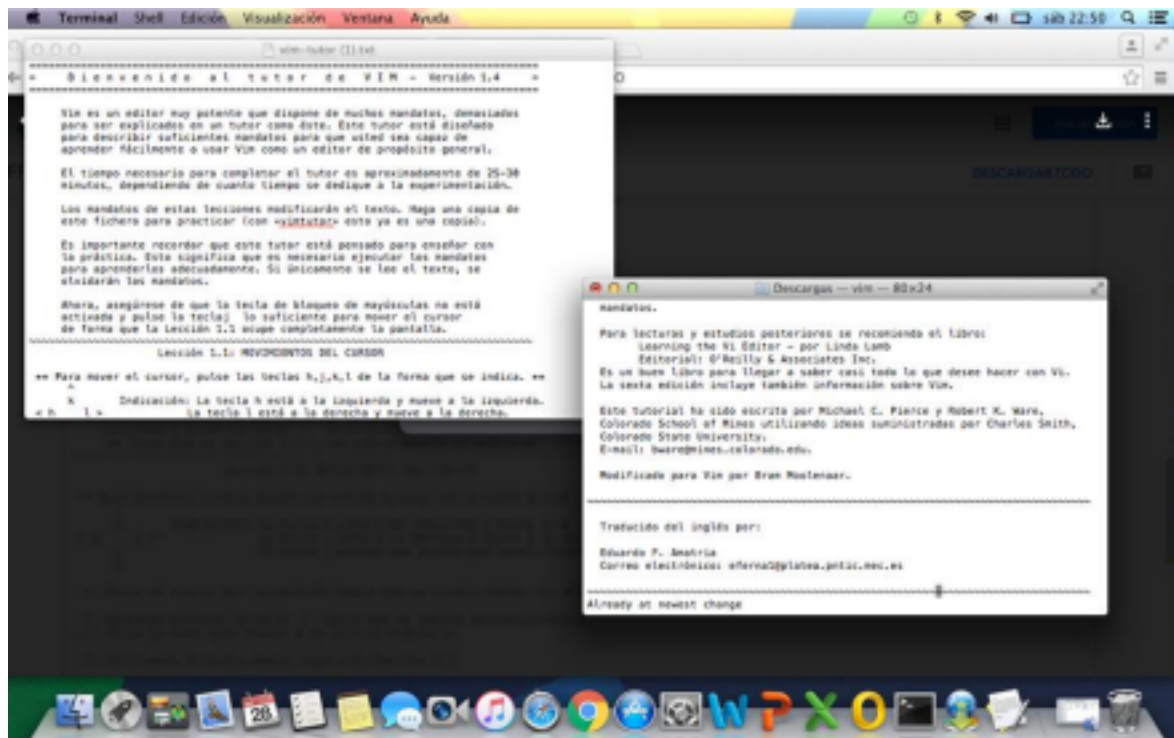
### Vi

Otro editor de terminal, aunque también tiene versiones gráficas.

Este es uno de los editores más populares, con más características, y que los expertos aseguran que ofrecen mayor ganancia de velocidad para editar textos.

La razón es que este editor es bastante diferente a lo que están acostumbrados. Se requiere más tiempo para aprenderlo a usar y para dominarlo.

## Actividad 2



```
de forma que la Lección 1.1 ocupe completamente la pantalla.

~~~~~
Lección 1.1: MOVIMIENTOS DEL CURSOR

** Para mover el cursor, pulse las teclas h,j,k,l de la forma que se indica. **

      k      Indicación: La tecla h está a la izquierda y mueve a la izquierda.
    < h      La tecla l está 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 cómodo con él.

2. Mantenga pulsada la tecla j hasta que se repita «automágicamente».
   —> Ahora ya sabe como llegar a la lección siguiente.

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

Nota: Si alguna vez no está 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 también funcionan. Pero usando
hjkl podrá moverse mucho más rápido una vez que se acostumbre a ello.

~~~~~
Lección 1.2: ENTRANDO Y SALIENDO DE VIM

!! NOTA: Antes de ejecutar alguno de los pasos siguientes lee 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 3

~~~~~
```

### 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 luz.

5. Ahora que la línea está correcta, continúe con la Lección 1.4.

NOTA: A medida que vaya avanzando en este tutor no intente memorizar, aprenda practicando.

### Lección 1.4: EDICIÓN DE TEXTO - INSERCIÓN

== Estando en modo Normal pulse `i` para insertar texto. ==  
`i` hola mundo!

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.

---> Fita 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 1

1. El cursor se mueve utilizando las teclas de las flechas o las teclas `h` `j` `k` `l`.  
`h` (izquierda) `j` (abajo) `k` (arriba) `l` (derecha)
2. Para acceder a Vim (desde el símbolo del sistema `%`) escriba:  
`vim FILENAME <INTR>`
3. Para salir de Vim escriba: `<ESC>` o `q!` `<INTR>` 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.

### 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.



#### Lección 2.2: MÁS MANDATOS PARA BORRAR

== Escribe dd 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. Escribe dd para borrar hasta el final de la línea.
- > Alguien ha escrito el final de esta línea dos veces.

#### Lección 2.3: SOBRE MANDATOS Y OBJETOS

El formato del mandato de borrar dd es como sigue:

[número] d objeto d 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 (línea, abajo).

Sea 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.

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

== Escribe 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 des des en una fila para borrar una línea.

1. Mueva el cursor a la segunda línea de la lista de abajo.
  2. Escribe dd para borrar la línea.
  3. Muevase ahora a la cuarta línea.
  4. Escribe ddd (recuerde número-mandato-objeto) para borrar las dos líneas.
- 1) Las rosas son rojas.  
2) El cielo es azul.  
3) El azúcar es dulce.  
4) Y así eres tú.

#### 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 sitíelo 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

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: **ds**
3. Para borrar una línea enter pulse: **dd**
4. El formato de un mandato en modo Normal es:  

[número]	mandato	objeto	@	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**

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

- \*\* Pulse **p** para poner lo último que ha borrado después del cursor. \*\***
1. Mueva el cursor al final de la lista de abajo.
  2. Escriba **dd** para borrar la línea y almacenarla en el buffer de VSE.
  3. Mueva el cursor a la línea que debe quedar por debajo de la línea a mover.
  4. Estando en mod Normal, pulse **p** para restituir la línea borrada.
  5. Repita los pasos 2 a 4 para poner todas las líneas en el orden correcto.
    - a) Las rosas son rojas,
    - b) Las violetas son azules,
    - c) La inteligencia se aprende,
    - d) ¿Puedes aprenderla tu?

### 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.

MacBook Pro

### 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!

### Lección 3.3: EL MANDATO «CHANGE» (cambiar)

- \*\* Para cambiar parte de una palabra o toda ella escriba **cw** . \*\***
1. Mueva el cursor a la primera línea de abajo señalada con **--->**.
  2. Sitúe el cursor en la **u** de lubrs.
  3. Escriba **cw** y corrija la palabra (en este caso, escriba 'linea').
  4. Pulse **<ESC>** y mueva el cursor al error siguiente (el primer carácter que debe 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: MÁS CAMBIOS USANDO «

» 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\$.

#### RESUMEN DE LA LECCIÓN 3

1. Para sustituir texto que ha sido borrado, pulse g. Esto pone el texto borrado DESPUÉS del cursor (si la 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.

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

» Pulse CTRL-g para mostrar su situación en el fichero y su estado.

Pulse MAYÚ-G para moverse a una determinada línea del fichero. »

Nota: Vea 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 MAYÚ-G para ir al final del fichero.

3. Escriba el número de la línea en la que estaba y después MAYÚ-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 este ejercicio 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, la misma que el mandato :.

2. Escriba ahora ~~erroror~~ «INTRO». Esta es la palabra que quiere buscar.

3. Para repetir la búsqueda, simplemente pulse s.

Para buscar la misma frase en la dirección opuesta, pulse MAYÚ-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.

~~erroror~~ no es la forma de deletrear error; ~~erroror~~ es un error.

#### Lección 4.3: BÚSQUEDA PARA COMPROBAR PARÉNTESIS

» Pulse % para encontrar el paréntesis correspondiente a ), [ o ] . »

1. Sitúe el cursor en cualquiera de los caracteres ), [ o ] en la línea de abajo señalada con —>.

2. Pulse ahora el carácter %.

#### LECCIÓN 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS

\*\* Pulse **h** 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 **h** .
3. El cursor debería situarse en el paréntesis **{**, corchete **[** o llave **(** correspondiente.
4. Pulse **h** para mover de nuevo el cursor al paréntesis, corchete o llave correspondiente.

→ Esta **f** es una línea de prueba con **{**, **[**, **(**, **}**, **]**, **)** en ella. **!D-** Nota: Esto es muy útil en la detección de errores en un programa con paréntesis, corchetes o llaves desparejados.

#### LECCIÓN 4.4: UNA FORMA DE CAMBIAR PALABRAS

\*\* Escriba **:/viejo/nuevo/** para sustituir 'viejo' por 'nuevo'. \*\*

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

1. Escriba **:/viejo/nuevo/** «INTRO». Tenga en cuenta que este mandato cambia sólo la primera aparición en la línea de la expresión a cambiar.
- Las mejores flores para ver las flores son las primaverales.
4. Para cambiar todas las apariciones de una expresión entre dos líneas escriba **:/#s/viejo/nuevo/** donde **#s** son los números de las dos líneas. Escriba **:/%s/viejo/nuevo/** para hacer los cambios en todo el fichero.

#### RESUMEN DE LA LECCIÓN 4

1. **Ctrl-g** muestra la posición del cursor en el fichero y su estado.  
**Mayú-G** mueve el cursor al final del fichero. Un número de línea seguido de **Mayú-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 **h** cuando el cursor está sobre **{**, **[**, **(**, **}**, **]**, **)** 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**

#### Lección 5.1: CÓMO 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».



### 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.

### 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 TAMBIÉN ESTE NÚMERO!
4. Para guardar SÓLO 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 está ahí con :!dir pero NO lo borre.

### Lección 5.4: RECUPERANDO Y MEZCLANDO FICHEROS

\*\* Para insertar el contenido de un fichero escriba :r NOMBRE\_DEL\_FICHERO \*\*

1. Escriba :!dir para asegurarse de que su fichero TEST del ejercicio anterior está presente.
2. Situe el cursor al principio de esta pantalla.

NOTA: Después de ejecutar el paso 3 se verá la Lección 5.3. Luego muévase hacia ABAJO para ver esta lección de nuevo.

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.

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.

### 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.

#### 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 Mayú-O cuando el curso esté en esta línea.

#### 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.

#### 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.

#### 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



---

## 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`

---

## 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>
```

---

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.

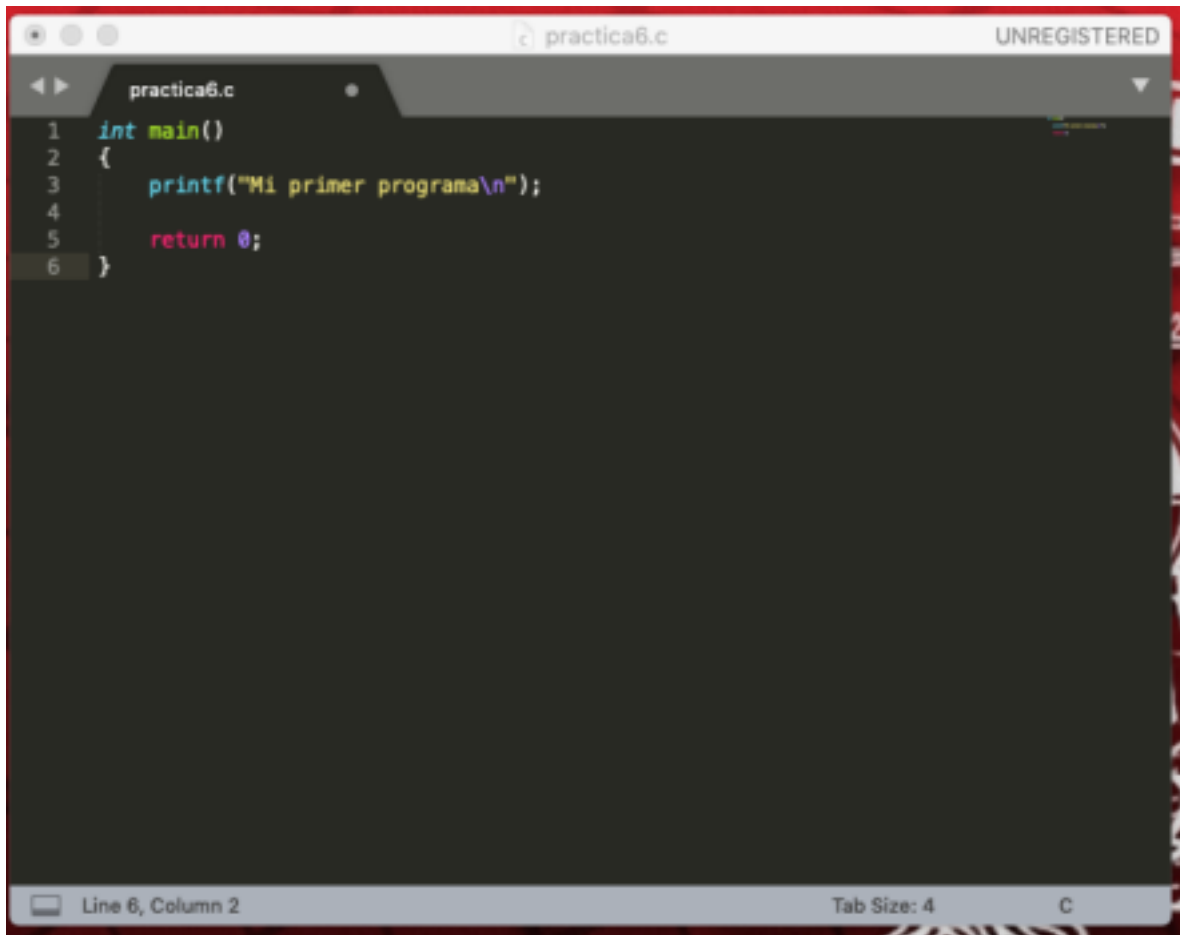
Para lecturas y estudios posteriores se recomienda el libro:  
Learning the Vi Editor - por Linda Lamb  
Editorial: O'Reilly & Associates Inc.



### Actividad 3

Notepad++

Es un editor gráfico mucho más parecido a lo que están acostumbrados.



The image shows a screenshot of the Notepad++ text editor. The window title bar at the top indicates the file is ' practica6.c' and the application is 'UNREGISTERED'. The editor has a dark theme. A single tab labeled ' practica6.c' is open. The code is as follows:

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

The status bar at the bottom shows 'Line 6, Column 2', 'Tab Size: 4', and the encoding 'C'.

```
fp03alu19 — -bash — 80x24
[Mauritania18:~ fp03alu19$ ls
Desktop      Downloads    Movies       Pictures     int main ().c
Documents    Library      Music        Public       practica6.c
[Mauritania18:~ fp03alu19$ gcc practica6.c -o areli
practica6.c:3:2: warning: implicitly declaring library function 'printf' with
      type 'int (const char *, ...)' [-Wimplicit-function-declaration]
      printf("Mi primer programa\n");
      ^
practica6.c:3:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
[Mauritania18:~ fp03alu19$ ./areli
Mi primer programa
[Mauritania18:~ fp03alu19$ ]
```

## Conclusiones:

En la parte del tutor de Vim me gustó mucho el tutorial porque está muy bien explicado y pude entender bien cómo es que se usa pues iba leyendo y haciendo lo que te iba indicando el manual lo que me hizo entender aún mejor pues al mismo tiempo iba practicando.

Las únicas lecciones que no puede llevar a cabo bien fueron la 4 y la 7, de hecho en la lección intente hacerlo y solo moví el texto y ya no puede regresarlo como estaba antes.

Mis lecciones favoritas fueron las de remplazo y en las que servían para borrar palabras líneas y ciertas partes de una oración.

La última actividad la repetí tres veces porque guarde dos veces lo del editor de texto (sublime tex) con diferentes nombres y al parecer eso era lo que no me dejaba vincular, pero al final lo logre como se muestra en las capturas.