

Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

<i>Profesor:</i>	Alejandro Pimentel
<i>Asignatura:</i>	Fundamentos de programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	6
<i>Integrante(s):</i>	Uno Karin Natalia
<i>No. de Equipo de cómputo empleado:</i>	26
<i>No. de Lista o Brigada:</i>	8723 #50
<i>Semestre:</i>	1
<i>Fecha de entrega:</i>	30 de septiembre del 2019
<i>Observaciones:</i>	

CALIFICACIÓN: _____

PRÁCTICA 6

Entorno de C

OBJETIVO

En esta práctica se verán las herramientas que se utilizarán para ejecutar el programa en Lenguaje C.

INTRODUCCIÓN

Empecemos definiendo qué es un programa. Un programa puede entenderse como la forma de expresar la solución a un problema de manera que sea comprensible para el ordenador. En otras palabras, un programa es un conjunto ordenado de instrucciones que se dan a la computadora indicando el conjunto de operaciones o tareas que se desea llevar a cabo. También, qué es el lenguaje de programación en C. El lenguaje de programación C es un lenguaje de alto nivel que se puede caracterizar por los siguientes puntos:

- Es de propósito general, esto significa que puede ser usado tanto para el desarrollo de sistemas operativos como para programas científicos, programas de aplicación o programas de educación y juegos.
- Posee una alta transportabilidad ya que los programas escritos en C pueden ser llevados de un tipo de ordenador a otro y funcionarán. Los cambios que hay que realizar son mínimos.
- Como solo tiene 32 palabras reservadas se dice que es compacto.
- Los compiladores generan ejecutables más pequeños porque por norma general, los programas escritos en lenguaje C poseen menos líneas de código que otros lenguajes de programación de alto nivel.
- El lenguaje de programación C es un lenguaje imperativo o procedimental. Esto significa que indica secuencias de acciones con el fin de llegar a un objetivo. Generalmente los lenguajes imperativos o procedimentales contienen una parte declarativa. El lenguaje C da órdenes a la máquina.
- Es un lenguaje estructurado ya que los programas escritos con él se pueden organizar en módulos.

Extraído de: <http://recursostic.educacion.es/observatorio/web/ca/software/programacion/745-introduccion-a-la-programacion-con-el-lenguaje-c>

DESARROLLO/RESULTADOS

Actividad 1.

Un Texto plano (plain text), son aquellos archivos formados exclusivamente por texto (sólo caracteres), sin ningún formato; es decir, no requieren ser interpretados para leerse (aunque pueden ser procesados en algunos casos). También son llamados archivos de texto llano, simple o sin formato. En otras palabras, son archivos que contienen solo texto, pero no hay información sobre el tipo de letra, ni formas (negrita, subrayados...), ni tamaños.

Tipos de archivos:

- Txt

La extensión **TXT** representa "textfile" (archivo de texto), que sustituyó a su antiguo nombre "flatfile" (archivo sin formato). Este archivo informático estructura series de líneas de texto.

Recopilado en: <https://www.online-convert.com/es/formato-de-archivo/txt>

- markdown

Markdown es una forma sencilla de agregar formato a textos web. Con formato nos referimos a itálicas, negritas, listas, y más, en texto plano –por ejemplo, el texto que nos encontramos en el visor de HTML de los editores de blogs, o en el Bloc de Notas de Windows, para hacer el concepto más asequible.

Recopilado en: <https://hipertextual.com/archivo/2013/04/que-es-markdown/>

- html

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto”

Recopilado en: <https://codigofacilito.com/articulos/que-es-html>

- LaTeX

LaTeX es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica.

Recopilado en: <https://es.wikipedia.org/wiki/LaTeX>

- csv

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.

Recopilado en: https://techlandia.com/archivo-extension-csv-hechos_47208/

EDITOR DE TEXTO. Son programas que permiten crear y modificar archivos digitales compuestos únicamente por texto sin formato, conocidos comúnmente como archivos de texto o texto plano. El [programa](#) lee el archivo e interpreta los bytes leídos según el código de caracteres que usa el editor. Hoy en día es comúnmente de 7- ó 8-bits en ASCII o UTF-8, rara vez EBCDIC.

Recopilado en: https://www.ecured.cu/Editor_de_texto

Tipos de editores de texto:

- Atom
- Bluefish
- Brackets
- Gedit
- Geany
- Emacs

- Nano
- Notepad++
- Pico
- Sublime Text
- Vim

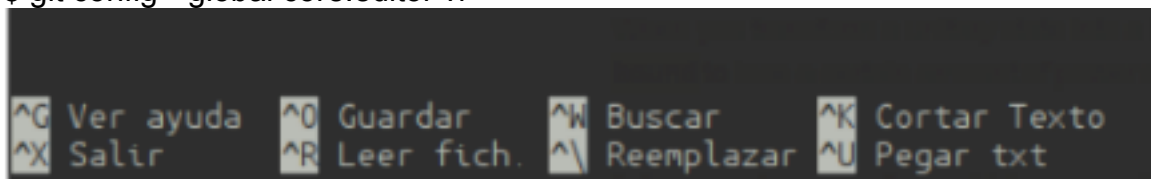
Varían en su función como resaltado de palabras clave, autocompletado, lista de elementos definidos, auto sangrado, identificación de pares de paréntesis, integración de compilador, integración de control de versiones, integración de terminal, búsquedas avanzadas, entre otras.

Nano

Es un editor de texto de terminal; ejecutando comandos sencillos, cortos y básicos, como:

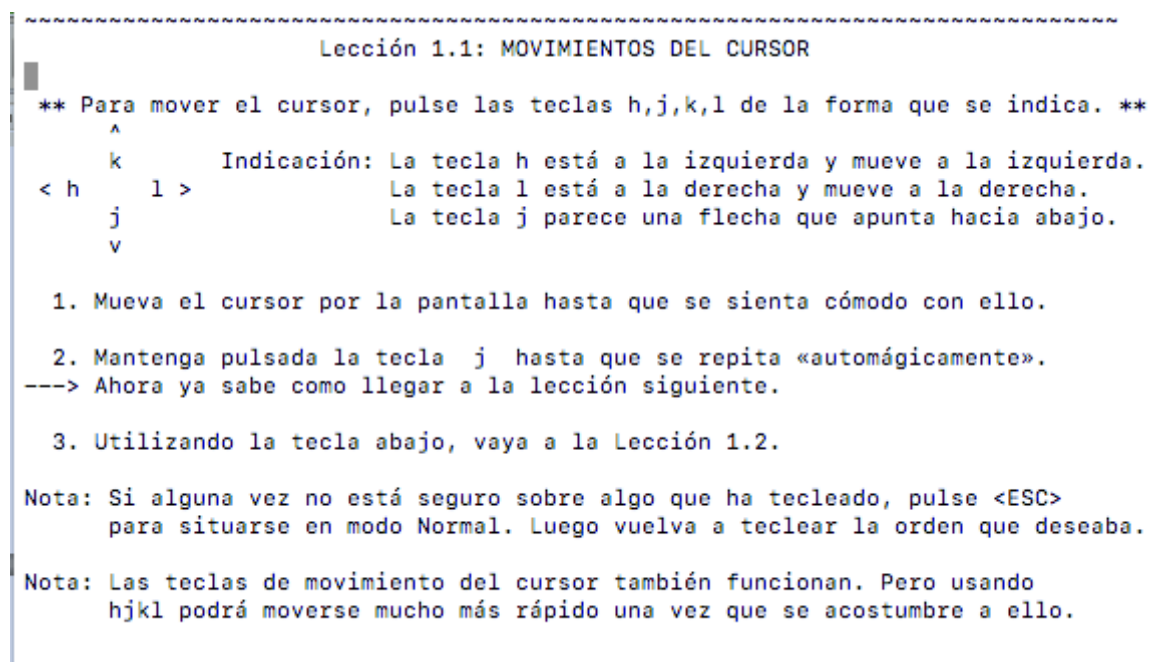
```
$ git config --global core.editor nano
```

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



Así como nano existe vi o vim es un editor de texto de terminal que es usado comúnmente por ser más rápido, aunque se requiere dominarlo para ser usado ya que no se maneja de la misma manera al resto.

Actividad 2. Vi tutor. En las siguientes imágenes se muestra cómo se ejecuto y los resultados obtenidos en ---→ o señalando en el cursor.



Lección 1.2: ENTRANDO Y SALIENDO DE VIM

;; 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> :q! <INTRO>

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

 :wp <INTRO> vimtutor <INTRO> :wq <INTRO>

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

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

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

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.

Lección 1.4: EDICIÓN DE TEXTO - INSERCIÓN

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

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

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>
vim FILENAME <INTRO>
<ESC>:q! <INTRO> escriba el texto
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.

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 bien que no pertenecen papel esta frase.

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.

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.

■

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,
- 4) Yo tengo un coche,
- 7) Y así eres tu.

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.

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 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.
 - 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.
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!

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 `'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 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: M?~AS 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\$.

RESUMEN DE LA LECCI?~SN 3

1. Para sustituir texto que ha sido borrado, pulse p . Esto pone el texto borrado DESPU?~IS 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.

Modificado para Vim por Bram Moolenaar.

Traducido del inglés por:

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

747,1 Final

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.

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 errorroor <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.

«errorroor no es la forma de deletrear error; ererrorroor 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 % .
 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.

Lección 4.4: UNA FORMA DE CAMBIAR ERRORES

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

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

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

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

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

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

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.

Lección 6.2: EL MANDATO «APPEND» (añadir)

**** Pulse a para insertar texto DESPU?~IS 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 DESPU?~IS 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» (remplazar)

**** 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

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»`

"w" and "W", "e" and "E" move to the start/end of the first word or WORD after a range of folded lines. "b" and "B" move to the start of the first word or WORD before the fold.

Special case: "cw" and "cW" are treated like "ce" and "cE" if the cursor is on a non-blank. This is because "cw" is interpreted as change-word, and a word does not include the following white space. {Vi: "cw" when on a blank followed by other blanks changes only the first blank; this is probably a bug, because "dw" deletes all the blanks}

Another special case: When using the "w" motion in combination with an operator and the last word moved over is at the end of a line, the end of that word becomes the end of the operated text, not the first word in the next line.

The original Vi implementation of "e" is buggy. For example, the "e" command will stop on the first character of a line if the previous line was empty. But when you use "2e" this does not happen. In Vim "ee" and "2e" are the same, which is more logical. However, this causes a small incompatibility between Vi and Vim.

```
=====
5. Text object motions                                object-motions

(                                                       (
[count] sentences backward.  exclusive motion.
)
[count] sentences forward.  exclusive motion.
```

motion.txt [Ayuda][R0]

414,69

~~~~~ Lección 7: MANDATOS PARA LA AYUDA EN L?~MNEAh

**** Utilice el sistema de ayuda en línea ****

Vim dispone de un sistema de ayuda en línea. Para activarlo, pruebe una de estas tres formas:

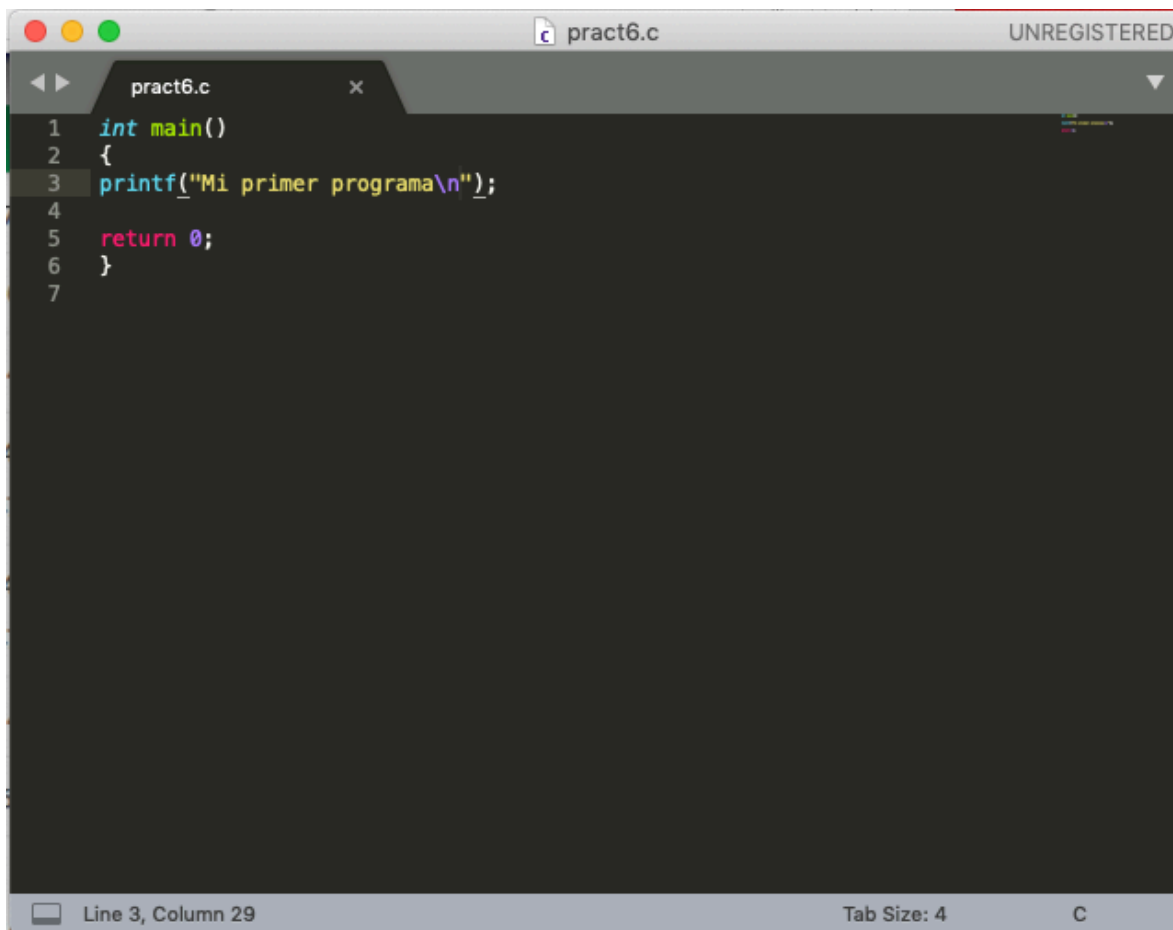
- pulse la tecla <AYUDA> (si dispone de ella)
- pulse la tecla <F1> (si dispone de ella)
- escriba :help <INTRO>

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>
```

Actividad 3. Notepad++



The image shows a screenshot of the Notepad++ text editor. The window title bar at the top indicates the file is 'pract6.c' and the application is 'UNREGISTERED'. The editor has a dark theme. A single tab labeled 'pract6.c' is open. The code is as follows:

```
1  int main()
2  {
3  printf("Mi primer programa\n");
4
5  return 0;
6  }
7
```

The status bar at the bottom shows 'Line 3, Column 29', 'Tab Size: 4', and the language is set to 'C'.

```
fp03alu50 — -bash — 80x24
)
^
1 warning and 6 errors generated.
[Nepal26:~ fp03alu50$ cd ]
[Nepal26:~ fp03alu50$ cd/ ]
-bash: cd/: No such file or directory
[Nepal26:~ fp03alu50$ gcc pract.6 -o pract.6_ ]
clang: error: unknown argument: '-o'
clang: error: no such file or directory: 'pract.6'
clang: error: no such file or directory: 'pract.6_'
clang: error: no input files
[Nepal26:~ fp03alu50$ gcc pract6.c -o pract6_ ]
pract6.c:3:1: warning: implicitly declaring library function 'printf' with type
'int (const char *, ...)' [-Wimplicit-function-declaration]
printf("Mi primer programa\n");
^
pract6.c:3:1: note: include the header <stdio.h> or explicitly provide a
declaration for 'printf'
1 warning generated.
[Nepal26:~ fp03alu50$ ./pract6_ ]
-bash: ./pract6_: No such file or directory
[Nepal26:~ fp03alu50$ ./pract6_ ]
Mi primer programa
Nepal26:~ fp03alu50$ █

fp03alu50 — -bash — 80x24
Last login: Mon Sep 23 09:21:58 on ttys000
[Nepal26:~ fp03alu50$ cd ]
[Nepal26:~ fp03alu50$ cd/ ]
-bash: cd/: No such file or directory
[Nepal26:~ fp03alu50$ ls ]
Desktop      Downloads    Movies      Pictures    pract6.c
Documents    Library     Music      Public     untitled
[Nepal26:~ fp03alu50$ gcc pract6.c -o pract6_ ]
pract6.c:3:8: error: expected parameter declarator
printf("Mi primer programa\n");
^
pract6.c:3:8: error: expected ')'
pract6.c:3:7: note: to match this '('
printf("Mi primer programa\n");
^
pract6.c:3:1: warning: type specifier missing, defaults to 'int'
[-Wimplicit-int]
printf("Mi primer programa\n");
^
pract6.c:3:31: error: expected ')'
printf("Mi primer programa\n");
^
pract6.c:2:1: note: to match this '('
(

```

CONCLUSIÓN

Podemos concluir que los programas de lenguaje en C son una forma más práctica de ejecutar programas utilizando solo el mismo teclado, sin el uso del

mouse o quitar las manos del teclado. También, los editores del texto son bastante útiles para los textos planos, ya que estos no necesitan un formato.