



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de programación

Grupo: 135

No de Práctica(s): 6

Integrante(s): Páez Matínez Karen

*No. de Equipo de
cómputo empleado:* 2

No. de Lista o Brigada: 3781

Semestre: 1

Fecha de entrega: 30/09/2019

Observaciones:

CALIFICACIÓN:

OBJETIVO:

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

1. 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. 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.
- **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. En esta lección nos centraremos en la primera acepción y aprenderemos a escribir archivos utilizando la sintaxis de Markdown.
- **Html:**
HTML, siglas en inglés de HyperText Markup Language ('lenguaje de marcas de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del Word Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la Word Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

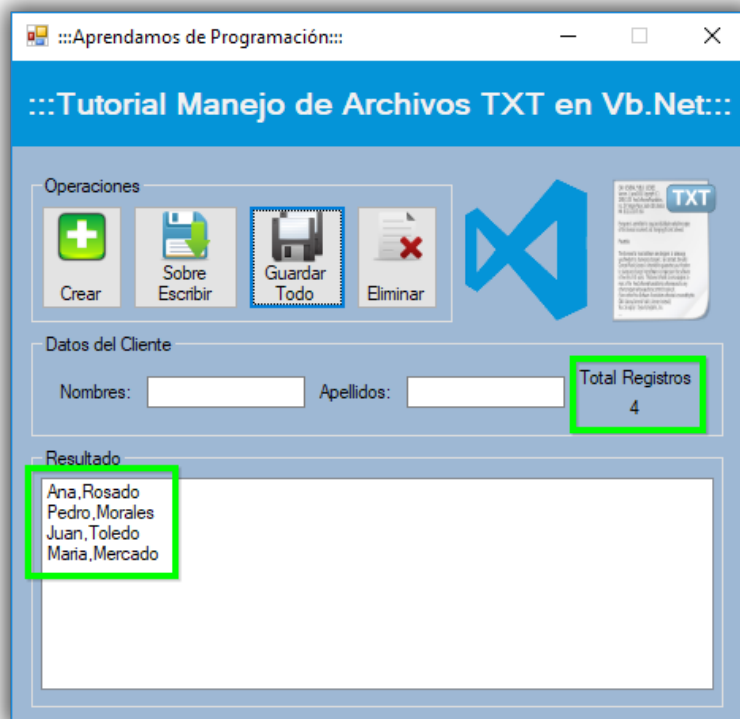
- **LaTeX:**

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.

- **Csv:**

Un csv (comma-separated values) es un archivo de texto que almacena los datos en forma de columnas, separadas por coma y las filas se distinguen por saltos de línea. Es una forma muy sencilla de representar la información.



ACTIVIDAD#1"Vim":

1. Primero descargamos el programa después empezamos seguir los pasos del tutorial de Vim:

```

** 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 en
una fila para borrar una línea.

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

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

```

Lección 2.5: EL MANDATO DESHACER

```

vim-tutor.txt[*] [unix] (12:54 29/09/2019) 224,7 27

```

```

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

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

---> Corrija los errores de esta línea y vuelva a ponerlos con deshacer.

8. Estos mandatos son muy útiles. Ahora pase al resumen de la Lección 2.

```

RESUMEN DE LA LECCIÓN 2

```

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

```

```

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!

```

- Después ubicar la carpeta en donde descargamos el tutorial de Vim después de esto vamos a ingresar con vi y con el nombre del archivo.

```
ACERREQUIPO MINGW64 ~ (master)
$ cd Downloads/
ACERREQUIPO MINGW64 ~/Downloads (master)
$ vi vim-tutor.txt
```

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

1. Pulse la tecla <ESC> (para asegurarse de que está en modo Normal).
2. Escriba:      :q! <INTRO>
--> Esto provoca la salida del editor SIN guardar ningún cambio que se haya
hecho. Si quiere guardar los cambios y salir escriba:
      :wq <INTRO>

3. Cuando vea el símbolo del sistema, escriba el mandato que le trajo a este
tutor. Este puede haber sido:  vimtutor <INTRO>
Normalmente se usaría:      vim tutor <INTRO>
--> 'vim' significa entrar al editor, 'tutor' es el fichero a editar.

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

vim-tutor.txt (unix) (12:54 29/09/2019)      54,0-1 60
~
```

- Después seguimos siguiendo en tutorial con los diferentes pasos:

```
MINGW64/Users/ACER/Downloads
Lección 1.3: EDICION DE TEXTO - BORRADO

** Estando en modo Normal pulse x para borrar el caracter 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 vaza saltó sobre la luna.
--> Falta algo de texto en esta línea.
--> Falta algo de texto en esta línea.
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.

vim-tutor.txt (unix) (12:54 29/09/2019)      46,0-1 54
```

```
MINGW64/Users/ACER/Downloads
Lección 1.4: EDICION DE TEXTO - INGERCION

** 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.

vim-tutor.txt (unix) (12:54 29/09/2019)      109,39-38 129
```

```
MINGW64/Users/ACER/Downloads
** Escriba de 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 dese borrar.
4. Pulse dw para hacer que la palabra desaparezca.

NOTA: Las letras de aparecerán en la última línea de la pantalla cuando
las escriba. Si escribe algo equivocado pulse <ESC> y continúe de nuevo.
--> Hay algunas palabras que no pertenecen a esta frase.

Lección 2.2: MAS MANDATOS PARA BORRAR

vim-tutor.txt (unix) (12:54 29/09/2019)      159,48 199
```

```
MINGW64/Users/ACER/Downloads
Lección 2.2: MAS MANDATOS PARA BORRAR

** Escriba ds para borrar hasta el final de la línea. **

1. Pulse <ESC> para asegurarse de que está en el modo Normal.
2. Mueva el cursor a la línea de abajo señalada con -->.
3. Mueva el cursor al final de la línea correcta (DESPUES del primer . ).
4. Escriba ds para borrar hasta el final de la línea.
--> Alguien ha escrito el final de esta línea dos veces.

vim-tutor.txt (unix) (12:54 29/09/2019)      177,59-58 219
```

<pre> ** 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 flla 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. Muevase ahora a la cuarta línea. 4. Escriba 2dd (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 2.1: EL MANDATO DESHACER vim-tutor.txt[*] [unix] (12:14 29/09/2019) 224,7 22 </pre>	<pre> ** Pulse u para deshacer los últimos mandatos, u para deshacer una línea entera. ** 1. Mueva el cursor a la línea de abajo señalada con --> y sitúelo bajo el primer error. 2. Pulse x para borrar el primer carácter erróneo. 3. Pulse ahora u para deshacer el último mandato ejecutado. 4. Ahora corrija todos los errores de la línea usando el mandato x. 5. Pulse ahora U mayúscula para devolver la línea a su estado original. 6. Pulse ahora u unas pocas veces para deshacer lo hecho por U y los mandatos previos. 7. Ahora pulse CTRL-R (mantenga pulsada la tecla CTRL y pulse R) unas pocas veces para volver a ejecutar los mandatos (deshacer lo deshecho). --> Corrija los errores de esta línea y vuelva a ponerlos con deshacer. 8. Estos mandatos son muy útiles. Ahora pase al resumen de la Lección 2. ----- RESUMEN DE LA LECCIÓN 2 vim-tutor.txt[*] [unix] (12:14 29/09/2019) 249,62-62 309 </pre>
<pre> ** Pulse u para deshacer los últimos mandatos, u para deshacer una línea entera. ** 1. Mueva el cursor a la línea de abajo señalada con --> y sitúelo bajo el primer error. 2. Pulse x para borrar el primer carácter erróneo. 3. Pulse ahora u para deshacer el último mandato ejecutado. 4. Ahora corrija todos los errores de la línea usando el mandato x. 5. Pulse ahora U mayúscula para devolver la línea a su estado original. 6. Pulse ahora u unas pocas veces para deshacer lo hecho por U y los mandatos previos. 7. Ahora pulse CTRL-R (mantenga pulsada la tecla CTRL y pulse R) unas pocas veces para volver a ejecutar los mandatos (deshacer lo deshecho). --> Corrija los errores de esta línea y vuelva a ponerlos con deshacer. 8. Estos mandatos son muy útiles. Ahora pase al resumen de la Lección 2. ----- RESUMEN DE LA LECCIÓN 2 vim-tutor.txt[*] [unix] (12:14 29/09/2019) 249,9 309 </pre>	<pre> línea a mover. 4. Estando en mod Normal, pulse o 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 tú? ----- 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. </pre>
<pre> 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 el 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 3.4: MAS CAMBIOS USANDO c ** El mandato change se utiliza con los mismos objetos que delete. ** vim-tutor.txt[*] [unix] (12:14 29/09/2019) 337,63-63 439 </pre>	<pre> ** 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 lubra. 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 deba cambiarse). 5. Repita los pasos 3 y 4 hasta que la primera frase sea igual a la segunda. --> Esta línea tiene unas pocas palabras que corregir usando el mandato change. --> Esta línea tiene unas pocas palabras que corregir usando el mandato change. ----- Lección 3.4: MAS CAMBIOS USANDO c ** El mandato change se utiliza con los mismos objetos que delete. ** vim-tutor.txt[*] [unix] (12:14 29/09/2019) 337,63-63 439 </pre>
<pre> ** 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 lubra. 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 deba cambiarse). 5. Repita los pasos 3 y 4 hasta que la primera frase sea igual a la segunda. --> Esta línea tiene unas pocas palabras que corregir usando el mandato change. --> Esta línea tiene unas pocas palabras que corregir usando el mandato change. ----- Lección 3.4: MAS CAMBIOS USANDO c ** El mandato change se utiliza con los mismos objetos que delete. ** vim-tutor.txt[*] [unix] (12:14 29/09/2019) 337,63-63 439 </pre>	<pre> 1. El mandato change funciona de la misma forma que delete. El formato es: [número] c [objeto] o c [número] objeto 2. Los objetos son también los mismos, tales como w (palabra), \$ (fin de la línea), etc. 3. Mueva el cursor a la primera línea de abajo señalada con -->. 4. Mueva el cursor al primer error. 5. Escriba c\$ para hacer que el resto de la línea sea como la segunda y pulse «ESC». --> El final de esta línea necesita ser corregido usando el mandato c\$. --> El final de esta línea necesita ser corregido usando el mandato c\$. ----- RESUMEN DE LA LECCIÓN 3 vim-tutor.txt[*] [unix] (12:14 29/09/2019) 360,74-73 468 </pre>

4. En esta parte vemos como esta el fichero y su situasion como lo dice el tutorial de Vim.


```

** 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 tubos.
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 deba cambiarse).
5. Repita los pasos 3 y 4 hasta que la primera frase sea igual a la segunda.

----> Esta línea tiene unas pocas palabras que corregir usando el mandato change.
----> Esta línea tiene unas pocas palabras que corregir usando el mandato change.

-----
Lección 3.4: MAS CAMBIOS USANDO c
-----
** El mandato change se utiliza con los mismos objetos que delete. **
vte-tutor.txt (*) (unix) (12:14 29/09/2019) 357,62-61 48%

```

```

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
-----
vte-tutor.txt (*) (unix) (12:14 29/09/2019) 360,74-73 41%

```

```

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 desea 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: efernald@platea.ontic.mec.es
vte-tutor.txt (*) (unix) (12:14 29/09/2019) 769,1 Píxeles

```

```

Pase ahora a la lección siguiente.

-----
Lección 4.1: SITUACION 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-g y pulse . 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 2.
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 un número, NO se mostrarán en la pantalla).
vte-tutor.txt (unix) (12:14 29/09/2019) 388,0-1 31%
vte-tutor.txt /68 líneas --50%--

```

5. Aquí se da cuenta de uno de como funciona para buscar las palabras:

```

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 <error> <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.

<error> no es la forma de deletrear error; error es un error.

vte-tutor.txt (unix) (12:14 29/09/2019) 427,02 54%

```

```

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 <error> <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.

<error> no es la forma de deletrear error; error es un error.

vte-tutor.txt (unix) (12:14 29/09/2019) 417,22 54%

```

6. Aquí seguimos lo que sigue después del funcionamiento de palabras:

```

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.

vte-tutor.txt (unix) (12:14 29/09/2019) 447,11 57%

```

```

Lección 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS
-----
** Pulse % para encontrar el paréntesis correspondiente a }, ] o } . **

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

----> Esto [ es una línea de prueba con (, [, ], [, y ] en ella. ]].
Nota: ¡Esto es muy útil en la detección de errores en un programa con
paréntesis, corchetes o llaves desaparejos.

vte-tutor.txt (unix) (12:14 29/09/2019) 447,44-43 57%

```

```

Lección 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS
-----
** Pulse % para encontrar el paréntesis correspondiente a }, ] o } . **

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

----> Esto [ es una línea de prueba con (, [, ], [, y ] en ella. ]].
Nota: ¡Esto es muy útil en la detección de errores en un programa con
paréntesis, corchetes o llaves desaparejos.

vte-tutor.txt (unix) (12:14 29/09/2019) 447,44-43 57%

```


7. Aquí vemos como sale de título de una forma de cambiar errores:

```
Lección 4.4: UNA FORMA DE CAMBIAR ERRORES

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

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

. 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.

. 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.

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

```
st_arm.exe
Lope+De++Fuenteovejuna.pdf
0180413_183315.3gp
0180418_184825.3gp
0180420_184653.3gp
0180427_185537.3gp
retirada (1).docx'
retirada.docx
tor.txt
NATO
al-dj-7.0 (1).exe'
al-dj-7.0.exe
al-dj-8-5-3573.msi
app Image 2018-10-17 at 9.41.18 PM.jpeg'
app Image 2019-01-13 at 7.06.50 PM (1).jpeg'
(1).docx'
docx
compat.diagcab
alifa - See You Again ft. Charlie Puth [Official Video] Furious 7 Soundtr
'
v-132.docx

INTRO o escriba una orden para continuar
```

8. Aqui también con ayuda del tutorial la lección 7 nos dice cómo escribir un comando con ayuda:

```
2. Repita varias veces la búsqueda pulsando la tecla n
3. Fije la opción 'to' (Ignorar la caja de la letra) escribiendo:
   :set to
4. Ahora busque 'ignore' de nuevo pulsando n
   Repita la búsqueda varias veces más pulsando la tecla n
5. Fije las opciones 'hlsearch' y 'nohsearch':
   :set hlsearch
   :set nohsearch
6. Ahora introduzca la orden de búsqueda otra vez, y vea qué pasa:
   /ignore

=====
RESUMEN DE LA LECCIÓN 6
=====
vim-tutor_481[4] (unix) (12/14 28/09/2019) 715,7 Bps
Ignore n
```

```
help.txt For vim version 8.1. Last change: 2019 Jul 21
VIM - main help file

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

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-].
With the mouse: Double-click the left mouse button on a tag, e.g. bars.
Jump back: Type CTRL-O. Repeat to go further back.

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

Normal mode command: :help
Normal mode command: :help s
help.txt (unix) (12/14 28/09/2019) 715,7 Bps
:help w <CTRL>
:help w <CTRL>
help.txt (unix) (12/14 28/09/2019) 715,7 Bps
```

En esta parte utilizamos la tecla F1

9. O también podemos ocupar el “help”

```
help.txt For vim version 8.1. Last change: 2019 Jul 21
VIM - main help file

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

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-].
With the mouse: Double-click the left mouse button on a tag, e.g. bars.
Jump back: Type CTRL-O. Repeat to go further back.

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

Normal mode command: :help
Normal mode command: :help s
help.txt (unix) (12/14 28/09/2019) 715,7 Bps
:help w <CTRL>
:help w <CTRL>
help.txt (unix) (12/14 28/09/2019) 715,7 Bps
```

10. Por ultimo como seguimos los ultimos pasos y ya:

```
W
:help w <CTRL>
:help w <CTRL>
help.txt (unix) (12/14 28/09/2019) 715,7 Bps
```

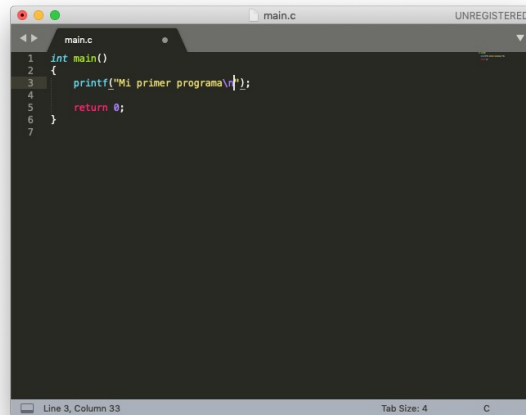
```
c_<T
:help w <CTRL>
:help w <CTRL>
help.txt (unix) (12/14 28/09/2019) 715,7 Bps
```

```
Insert-index
:help w <CTRL>
:help w <CTRL>
help.txt (unix) (12/14 28/09/2019) 715,7 Bps
```

Insert-index

“Notepad++”

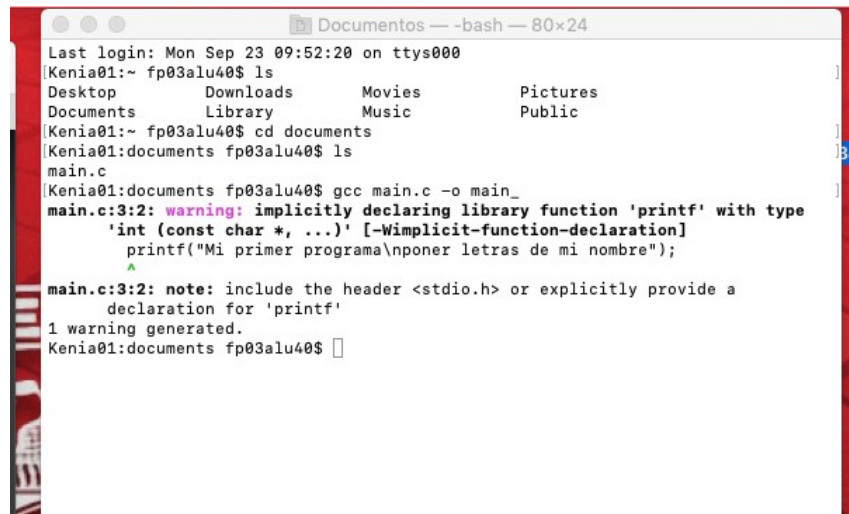
1. Aquí en la clase empezamos a realizar el ejemplo de cómo empezarlo a realizar en sublime text. Empezamos a seguir los pasos de la practica 6,y lo guardamos en un archivo en este caso lo guarde en una de documentos.



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

Line 3, Column 33 Tab Size: 4 C

2. Después de haberlo guardado en documentos abrimos el terminador, y escogemos la de documentos y de ahí vamos siguiendo los pasos de la práctica #6.



```
Documentos — -bash — 80x24
Last login: Mon Sep 23 09:52:20 on ttys000
Kenia01:~ fp03alu40$ ls
Desktop      Downloads    Movies       Pictures
Documents    Library      Music        Public
Kenia01:~ fp03alu40$ cd documents
Kenia01:documents fp03alu40$ ls
main.c
Kenia01:documents fp03alu40$ 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\nponer letras de mi nombre");
^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
declaration for 'printf'
1 warning generated.
Kenia01:documents fp03alu40$
```

```
Documents — -bash — 80x24
Last login: Mon Sep 23 09:52:20 on ttys000
Kenia01:~ fp03alu40$ ls
Desktop      Downloads    Movies       Pictures
Documents    Library      Music        Public
Kenia01:~ fp03alu40$ cd documents
Kenia01:documents fp03alu40$ ls
main.c
Kenia01:documents fp03alu40$ 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\nponer letras de mi nombre");
^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
declaration for 'printf'
1 warning generated.
Kenia01:documents fp03alu40$ ./main_
Mi primer programa
poner letras de mi nombreKenia01:documents fp03alu40$
```

```
Documents — -bash — 80x24
Last login: Mon Sep 23 09:54:45 on ttys000
Kenia01:~ fp03alu40$ ls
Desktop      Downloads    Movies       Pictures
Documents    Library      Music        Public
Kenia01:~ fp03alu40$ cd documents
Kenia01:documents fp03alu40$ 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\nponer letras de mi nombre\n");
^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
declaration for 'printf'
1 warning generated.
Kenia01:documents fp03alu40$ ./main_
Mi primer programa
poner letras de mi nombre
Kenia01:documents fp03alu40$
```

CONCLUSION:

En esta práctica se me dificultó un poco al tratar de bajar los programas, pero al realizarlos creo que fue más sencillo porque me gustó seguir las especificaciones del profesor fue algo interesante y divertido aprender cosas nuevas como que el significa un texto plano y que existe varios tipos de este y en diferentes archivos de texto plano como los que el profesor explico en la práctica #6, en lo particular siento que me salió bien el trabajo aun que me costó trabajo.