



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

<i>Profesor:</i>	M.C. Alejandro Esteban Pimentel Alarcon
<i>Asignatura:</i>	Fundamentos de programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	6
<i>Integrante(s):</i>	Yuan Xiaojing
<i>No. de Equipo de cómputo empleado:</i>	35
<i>No. de Lista o Brigada:</i>	317693612
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	26-08-19
<i>Observaciones:</i>	

CALIFICACIÓN: _____

Introducción

Un lenguaje C es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

● Actividad 1

1. Txt

La extensión .txt es un documento sin formato, es simplemente texto. En general, cualquier editor de texto permite editar un archivo .txt. El .txt es simple texto, no tiene ningún tipo de formato y ocupa muy poco.

2. Markdown

Markdown es un lenguaje de marcado ligero creado por John Gruber. El objetivo que busca este lenguaje es mejorar la facilidad de publicación de contenido. Es decir, que escribas más rápido y obtengas un resultado final mejor. Con simples comandos mientras estás escribiendo puedes darle formato al texto final que obtendrás.

3. html

HTML es un lenguaje de marcado de hipertexto o "HyperText Markup Language" por el desarrollo de sus iniciales en inglés, básicamente este lenguaje se escribe en su totalidad con elementos, estos elementos están constituidos por etiquetas, contenido y atributos, que explicaremos de una manera más detallada en algunas líneas más abajo.

HTML es un lenguaje que interpreta el navegador web para mostrar los sitios o aplicaciones web tal y como estamos acostumbrados.

4. LaTeX

LaTeX. Es un sistema de composición de textos que está orientado especialmente a la creación de documentos científicos que contengan formulas matemáticas, cuadros y tablas. Además, también se pueden crear otros tipos de documentos, que pueden ser desde cartas sencillas hasta libros completos. LATEX está organizado sobre TEX. LATEX está disponible para la mayoría de los miniordenadores y microordenadores, desde IBM PCs en adelante.

5. 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 2 vim

1. Abrir el archivo vim-tutor.txt que está en el drive
2. Abrir terminal y la ejecutar como administrador
3. Con cd llegar a la carpeta donde se guarda el archivo y abrir con vim vim-tutor.txt.

```

MINGW64:/c/Users/Yuan/downloads
=====
B i e n v e n i d o   a l   t u t o r   d e   V I M   -   V e r s i ó n   1 . 4
=====

Vim es un editor muy potente que dispone de muchos mandatos, demasiados
para ser explicados en un tutor como éste. Este tutor está diseñado
para describir suficientes mandatos para que usted sea capaz de
aprender fácilmente a usar Vim como un editor de propósito general.

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

Los mandatos de estas lecciones modificarán el texto. Haga una copia de
este fichero para practicar (con «vimtutor» esto ya es una copia).

Es importante recordar que este tutor está pensado para enseñar con
la práctica. Esto significa que es necesario ejecutar los mandatos
para aprenderlos adecuadamente. Si únicamente se lee el texto, se
olvidarán los mandatos.

Ahora, asegúrese de que la tecla de bloqueo de mayúsculas no está
activada y pulse la tecla j lo suficiente para mover el cursor
vim-tutor.txt [unix] (01:22 29/09/2019) 4,0-1 Comienzo

```

4. Seguir el tutor, intentar a mover con las flechas y las letras h, j, k y l

```

MINGW64:/c/Users/Yuan/downloads
Lección 1.1: MOVIMIENTOS DEL CURSOR

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

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

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.
vim-tutor.txt [unix] (01:22 29/09/2019) 30,8-11 3%

```

5. Pulsar <Esc> antes de salir el editor; con el comando :q! es salirse sin guardar los cambios

```

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] (01:22 29/09/2019) 55,16-32
:q!

```

Con el comando :wq es guardar los cambios y salirse

```

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] (15:50 29/09/2019) 57,1 6%
:wq

```

6. Usar comando vim para volver a entrar al editor

```
Yuan@LAPTOP-DM3JH23S MINGW64 ~/downloads
$ vim vim-tutor.txt
```

7. Para editar un texto hay que mover el cursor hasta que esté bajo el carácter que va a ser borrado y pulsar la tecla x para borrar el carácter sobrante.

```
5. 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 está correcta, continúe con la Lección 6.
```

8. Para insertar un texto hay que pulsar la tecla i y escribir los caracteres a añadir, a medida que sea corregido cada error pulse <ESC> para volver al modo normal.

```
4. A medida que sea corregido cada error pulse <ESC> para volver al modo
Normal. Repita los pasos 2 a 4 para corregir la línea.
---> Falta algo de texto en esta línea.
---> Falta algo de texto en esta línea.
5. Cuando se sienta cómodo insertando texto pase al modo inserta.
```

9. Para borrar una palabra primero hay que pulsar <ESC> para que esté en el modo normal, mover el cursor al comienzo de una palabra que desee borrar, y pulsar dw para eliminar esa palabra.

```
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 pásalo bien que no pertenecen a esta frase.
```

10. Para borrar una oración pulsar <ESC> para asegurar que está en el modo normal y mover el cursor antes que la oración que quieres borrar y escribir d\$ para borrar.

```
4. Escriba d$ para borrar hasta el final de la línea.
--> Alguien ha escrito el final de esta línea dos veces.
```

11. Para borrar una línea entera primero hay que mover el cursor antes que la línea y escribir dd.

```
4. Escriba 2dd (recuerde número-mandato-objeto) para borrar líneas.
1) Las rosas son rojas,
3) El cielo es azul,
4) Yo tengo un coche,
7) Y así eres tu.
```

12. EL MANDATO DESHACER

- 1) Mueva el cursor a la línea de abajo señalada con ---> y sitúelo bajo el primer error. Pulse x para borrar el primer carácter erróneo.

```
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.
```

- 2) Pulse ahora u para deshacer el último mandato ejecutado.

```
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.
```

- 3) Ahora corrija todos los errores de la línea usando el mandato x.

```
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.
```

- 4) Pulse ahora U mayúscula para devolver la línea a su estado original.

```
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.
```

- 5) Pulse ahora u unas pocas veces para deshacer lo hecho por U y los mandatos previos.

```
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.
```

- 6) Ahora pulse CTRL-R (mantenga pulsada la tecla CTRL y pulse R) unas pocas veces para volver a ejecutar los mandatos (deshacer lo deshecho).

```
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

13. El mandato put , después de usar dd para eliminar un alínea con la tecla p se pueda restituir la línea borrada.

```
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,
```

14. Para sustituir un carácter equivocado pulsar r y un carácter correcto.

```
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!
```

15. Para cambiar parte de una palabra o toda ella escriba cw .

```
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 **
```

16. El mandato change se utiliza con los mismos objetos que delete. El formato es [número] c objeto.

```
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
```

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

```
está situado. Recuerde el número de la línea para el Paso 3.
Pulse Mayu-G para ir al final del fichero.
tutor.txt[+] [unix] (15:50 29/09/2019) 396,11 50%
tutor.txt [Modificado] 769 lines --51%--
```

Pulse MAYU-G para moverse a una determinada línea del fichero.
Para volver a regresar a la línea que estaba hay que escribir el número de la línea y pulsar MAYU-G.

```
~~~~~
Traducido del inglés por:
Eduardo F. Amatria
Correo electrónico: efernal@platea.pntic.mec.es

~~~~~
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 769,1 Final
396
```

18. Para buscar algo hay que seguir el formado `:/` carácter `<INTRO>`.
Para repetir la búsqueda pulse `n`, para buscar la frase en la dirección opuesta pulse `MAYU-N`, y para buscar la frase en la dirección opuesta utilice el mandato `?` en lugar de `/`.

```
principio.  
«errroor» no es la forma de deletrear error; errroor es un error.  
~~~~~  
Lección 4.3: BÚSQUEDA PARA COMPROBAR PARÉNTESIS  
** Pulse % para encontrar el paréntesis correspondiente a ),] o }. **  
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 428,3 55%  
:/errroor
```

19. Pulsar `%` para encontrar el paréntesis correspondiente a `)`, `]` o `}`.

```
rueba con (, [, {, y } en ella.)).  
detección de errores en un programa con  
llaves desaparejos.  
~~~~~  
: UNA FORMA DE CAMBIAR ERRORES
```

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

```
---> Laas mejores épocas para ver laas flores son laas primaveras.  
4. Para cambiar todas las apariciones de una expresión ente dos líneas  
escriba :#,#s/viejo/nuevo/g donde #,# son los números de las dos  
líneas. Escriba :%s/viejo/nuevo/g para hacer los cambios en todo  
el fichero.  
vim-tutor.txt [unix] (15:50 29/09/2019) 464,36-  
:s/laas/las
```

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

```
--->Nota: De esta manera es posible ejecutar cualquier mandato externo.  
--->Nota: Todos los mandatos : deben finalizarse pulsando <INTRO>.  
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 514,0-1 66%  
:!
```

22. Para guardar los cambios hechos en un fichero, escriba `:w` `NOMBRE_DE_FICHERO`.

```
--->Nota: De esta manera es posible ejecutar cualquier mandato externo.  
--->Nota: Todos los mandatos : deben finalizarse pulsando <INTRO>.  
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 514,0-1 66%  
:w TEST
```

23. Para guardar parte del fichero escriba `:#,#` `NOMBRE_DEL_FICHERO`

```
** Para insertar el contenido de un fichero escriba :r NOMBRE_DEL_FICHERO **  
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 561,1 73%  
:543,563 w TEST
```

24. Para insertar el contenido de un fichero escriba :r
NOMBRE_DEL_FICHERO. Después ese paso se puede ver que hay
dos copias de la lección 5.3.

```
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.  
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 1149,1 74%  
:r TEST
```

25. Pulsar o para abrir una línea debajo del cursor y situarle en modo
Insert, con MAYU-O se inserta una línea por encima del cursor.

```
3. Ahora copie la línea señalada con --> y púlsese <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.  
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 2162,1 93%
```

26. Pulsar a para insertar texto DESPUÉS del cursor. A mayúscula
añade texto al final de la línea).

```
3. Complete ahora la primera línea. Nótese que append es exactamente lo  
mismo que modo Insert, excepto por el lugar donde se inserta el texto.  
--> Esta línea le permitirá practicar el añadido de texto al final de una línea.  
--> Esta línea le permitirá practicar el añadido de texto al final de una línea.  
~~~~~  
Lección 6.3: OTRA VERSIÓN DE «REPLACE» (reemplazar)  
** Pulse una R mayúscula para sustituir más de un carácter. **  
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 2188,84-80 95%
```

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

```
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 anterior use las teclaS.  
4. Nótese que cuando pulse <ESC> para salir, el texto no alterado permanece.
```

28. Fijar una opción de forma que una búsqueda o sustitución ignore la
caja

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

```
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  
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 2221,6 96%  
:/ignorar
```


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

```
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

vim-tutor.txt[+] [unix] (15:50 29/09/2019) 2024,6 8
:set ic
```

- 3) Ahora busque 'ignorar' de nuevo pulsando n y repita la búsqueda varias veces más pulsando la tecla n. Fije las opciones 'hlsearch' y 'insearch': :set hls is

```
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

vim-tutor.txt[+] [unix] (15:50 29/09/2019) 2026,20 87%
:set hls is
```

- 4) Ahora introduzca la orden de búsqueda otra vez, y vea qué pasa: /ignore

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

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

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

=====
RESUMEN DE LA LECCIÓN 6

vim-tutor.txt[+] [unix] (15:50 29/09/2019) 2033,6 88%
```

29. Utilice el sistema de ayuda en línea, usar el comando :help <INTRO>, para cerrar la ventana de ayuda usar el comando :q <INTRO>

```
MINGW64/c:/Users/Yuan/downloads

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

VIM - main help file

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

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

Jump back: Type CTRL-O. Repeat to go further back.

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

WHAT      PREPEND  EXAMPLE
Normal mode command      :help x

elp.txt[Ayuda][~][RO] [unix] (08:41 17/08/2019) 1,1 Comienzo
:help w <INTRO>
vim-tutor.txt[+] [unix] (15:50 29/09/2019) 2075,6 89%
help.txt" [Solo lectura][UNIX] 238L, 8894C
```

Conclusión

Vim es un editor de texto muy rápido, tiene varios mandatos para ayudarnos a editar un texto, pero una vez que se domina y sobre el que podemos construir nuestro propio IDE, mediante extensiones, que se adapte a nuestras necesidades. Lenguaje C es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores, tiene un nivel medio para aprenderlo, y es un lenguaje muy versátil para los programadores