



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 PROGRAMACION

Grupo:

03

No de Práctica(s):

PRACTICA No. 6

Integrante(s):

MORONES FLORES INGRID YOHUALLI.

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

Semestre:

PRIMER SEMESTRE.

Fecha de entrega: 30/09/2019

Observaciones:

CALIFICACIÓN: _____

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.

LENGUAJE C

Al igual que B, 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.

Se trata de un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

TEXTO PLANO.

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.

MARKDOWN.

Markdown es un lenguaje de marcado que *facilita la aplicación de formato* a un texto empleando una serie de caracteres de una forma especial. En principio, fue pensado para elaborar textos cuyo destino iba a ser la web con más rapidez y sencillez que si estuviésemos empleando directamente HTML. Y si bien ese suele ser el mejor uso que podemos darle, también podemos emplearlo para cualquier tipo de texto, independientemente de cual vaya a ser su destino.

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”.

No obstante, este tipo de definiciones no nos dice mucho porque la definición es técnica. Para algunas personas al leer esto, piensan que HTML incluye el diseño

gráfico de las páginas web, sin embargo, eso no es cierto ya que HTML sólo sirve para indicar como va ordenado el contenido de una página web. Esto lo hace por medio de las marcas de hipertexto las cuales son etiquetas conocidas en inglés como tags.

La TeX

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, TeX, 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.

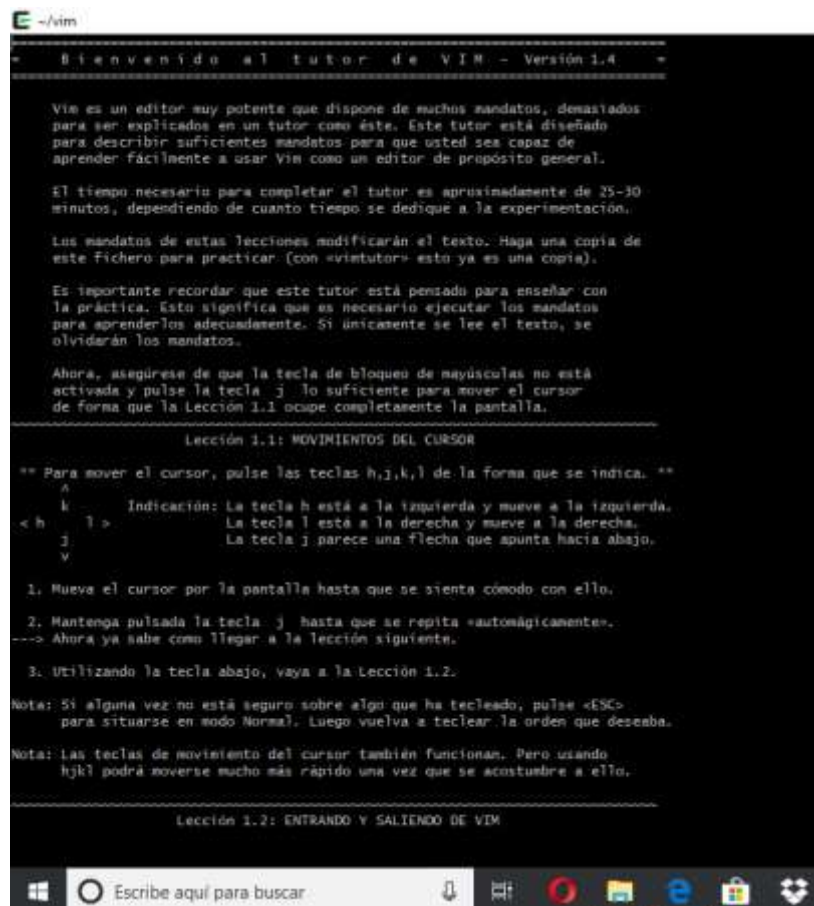
CSV

Los archivos **CSV** (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal).

VIM TUTOR.

En este caso fuimos siguiendo el archivo del tutor de Vim que se ve a continuación ara interactuar más con los distintos editores de texto que hay.

Para poder ejecutarlo gradamos el archivo en la carpeta de cygwin, posteriormente abrimos la terminal para que accediéramos al tutor y en el se iban dando una serie de pasos que el usuario tenia que seguir.



```
~vim
+-----+
+ Bienvenido al tutor de VIM - Versión 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
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. **
A
k - Indicación: La tecla h está a la izquierda y mueve a la izquierda.
< h l > 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 ello.
2. Mantenga pulsada la tecla j hasta que se repita «automáticamente».
---> 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
hjk l podrá moverse mucho más rápido una vez que se acostumbre a ello.

-----
Lección 1.2: ENTRANDO Y SALIENDO DE VIM
```

 ~/vim

```
p
hugo morones@LAPTOP-FGPEUFLH ~
$ pwd
/home/hugo morones

hugo morones@LAPTOP-FGPEUFLH ~
$ ls
vim

hugo morones@LAPTOP-FGPEUFLH ~
$ cd vim

hugo morones@LAPTOP-FGPEUFLH ~/vim
$ ls
vim-tutor.txt

hugo morones@LAPTOP-FGPEUFLH ~/vim
$ vi vim-tutor.txt

hugo morones@LAPTOP-FGPEUFLH ~/vim
$ vim tutor
-bash: vim: no se encontró la orden

hugo morones@LAPTOP-FGPEUFLH ~/vim
$ vi vim-tutor.txt

hugo morones@LAPTOP-FGPEUFLH ~/vim
$ vi vim|
```

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>`

---> 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.

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

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.

C

---> 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 LECCION 1

1. El cursor se mueve utilizando las teclas de las flechas o las teclas hjkl.
h (izquierda) j (abajo) k (arriba) l (derecha)
2. Para acceder a Vim (desde el símbolo del sistema %) escriba:
vim FILENAME <INTRO>
3. Para salir de Vim escriba: <ESC> :q! <INTRO> para eliminar todos los cambios.
4. Para borrar un carácter sobre el cursor en modo Normal pulse: x
5. Para insertar texto en la posición del cursor estando en modo Normal:
pulse i escriba el texto pulse <ESC>

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

Ahora continúe con la Lección 2.



 ~/vim

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

**** 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.|



Escribe aquí para buscar



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.
3. Muévase ahora a la cuarta línea.
4. Escriba `2dd` (recuerde número-mandato-objeto) para borrar las dos
 - 2) El barro es divertido,
 - 3) El cielo es azul,
 - 6) El azúcar es dulce,
 - 7) Y así eres tu.

Lección 2.5: EL MANDATO DESHACER

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

\$ - 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.
3. Muévase ahora a la cuarta línea.
4. Escriba 2dd (recuerde número-mandato-objeto) para borrar las dos
 - 2) El barro es divertido,
 - 3) El cielo es azul,
 - 6) El azúcar es dulce,
 - 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.

RESUMEN DE LA LECCIÓN 2

1 change; before #36 23:48:49



~/vim

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

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.

D

---> ¡Cuando esta línea fue escrita alguien pulso algunas teclas equivocadas!

---> ¡Cuando esta línea fue escrita alguien pulsó algunas teclas equivocadas!



○ Escribe aquí para buscar



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 'ínea').
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. ****

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

Lección 4.2: EL MANDATO «SEARCH» (buscar)

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

1. En modo Normal pulse el carácter /. Fíjese que tanto el carácter / como el cursor aparecen en la última línea de la pantalla, lo mismo que el mandato : .
2. Escriba ahora `errroor <INTRO>`. Esta es la palabra que quiere buscar.
3. Para repetir la búsqueda, simplemente pulse `n` .
Para busacar la misma frase en la dirección opuesta, pulse `Mayu-N` .
4. Si quiere buscar una frase en la dirección opuesta (hacia arriba), utilice el mandato `?` en lugar de `/` .

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

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

Lección 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS

**** Pulse % para encontrar el paréntesis correspondiente a),] o } . ****

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

---> Esto (es una línea de prueba con (, [,], {, y } en ella.))).

Nota: ¡Esto es muy útil en la detección de errores en un programa con paréntesis, corchetes o llaves disparejos.

Lección 4.4: UNA FORMA DE CAMBIAR ERRORES

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: MAS 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` ****

`:!ls`



Escribe aquí para buscar



~/.vim

```
hugo morones@LAPTOP-FGPEUFLH ~  
$ pwd  
/home/hugo morones  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ls  
vim  
hugo morones@LAPTOP-FGPEUFLH ~  
$ cd vim  
hugo morones@LAPTOP-FGPEUFLH ~/vim  
$ vi vim-tutor.txt  
[No write since last change]  
vim-tutor.txt  
Press ENTER or type command to continue  
[No write since last change]  
vim-tutor.txt  
Press ENTER or type command to continue  
[No write since last change]  
vim-tutor.txt  
Press ENTER or type command to continue  
TEST vim-tutor.txt  
Press ENTER or type command to continue
```

Lección 6.1: EL MANDATO «OPEN» (abrir)

**** Pulse o para abrir una línea debajo del cursor
y situarle en modo Insert ****

1. Mueva el cursor a la línea de abajo señalada con --->.
2. Pulse o (minúscula) para abrir una línea por DEBAJO del cursor y situarle en modo Insert.
3. Ahora copie la línea señalada con ---> y pulse <ESC> para salir del modo Insert.

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

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

4. Para abrir una línea por encima del cursor, simplemente pulse una O mayúscula, en lugar de una o minúscula. Pruebe este en la línea siguiente. Abra una línea sobre ésta pulsando Mayu-O cuando el curso está en esta línea.

Lección 6.2: EL MANDATO «APPEND» (añadir)

**** Pulse a para insertar texto DESPUES 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 DESPUES 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.

~/vim

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 DESPUES 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 DESPUES 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



~/vim

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.

Es un buen libro para llegar a saber casi todo lo que desee hacer con Vi. La sexta edición incluye también información sobre Vim.

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

Modificado para Vim por Bram Moolenaar.

Traducido del inglés por:

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



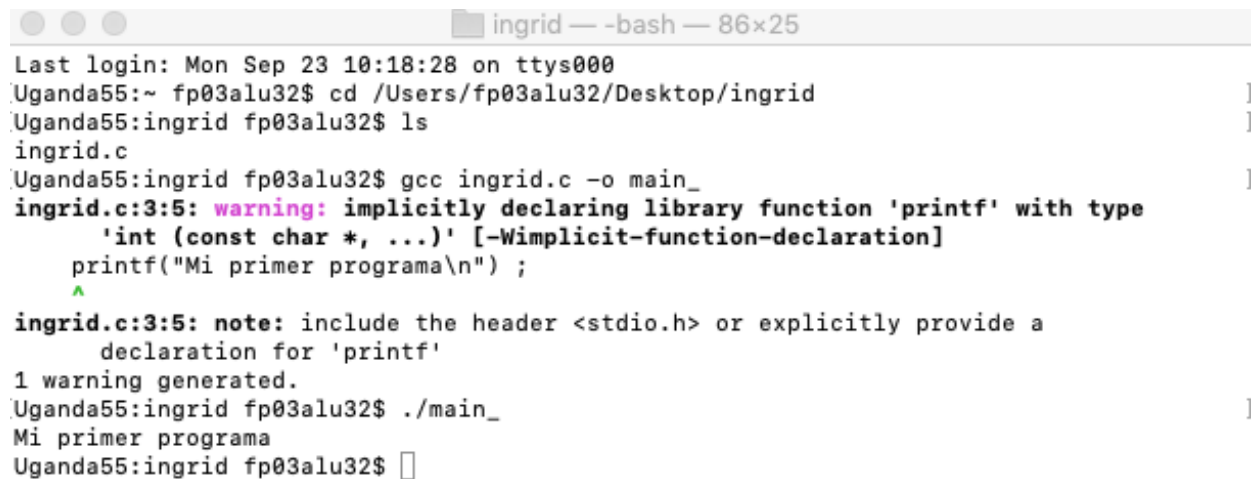
Escribe aquí para buscar






NOTEPAD ++

Por ultimo hicimos esta actividad en notepad++ la cual consistía en ejecutar nuestro primer programa siguiendo una serie de pasos que hacían que esto se llevara a cabo.



```
ingrid — -bash — 86x25
Last login: Mon Sep 23 10:18:28 on ttys000
Uganda55:~ fp03alu32$ cd /Users/fp03alu32/Desktop/ingrid
Uganda55:ingrid fp03alu32$ ls
ingrid.c
Uganda55:ingrid fp03alu32$ gcc ingrid.c -o main_
ingrid.c:3:5: warning: implicitly declaring library function 'printf' with type
      'int (const char *, ...)' [-Wimplicit-function-declaration]
      printf("Mi primer programa\n") ;
      ^
ingrid.c:3:5: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
Uganda55:ingrid fp03alu32$ ./main_
Mi primer programa
Uganda55:ingrid fp03alu32$
```



```
ingrid.c
UNREGISTERED

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

CONCLUSIONES.

El objetivo se cumplió, logramos ver los diferentes editores de texto así como practicar alguno de ellos en el caso de Vim y de Notepad ++.