



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Alejandro E. Pimentel Alarcón

Profesor:

Fundamentos de programación

Asignatura:

3; Bloque 135

Grupo:

6

No de Práctica(s):

Bazaldúa Morales Vanessa

Integrante(s):

*No. de Equipo de
cómputo*

Tailandia 47

#6

No. de Lista o

2020-1

Semestre:

02/Oct/19

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Entorno 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.

ACERCA DEL LENGUAJE C

C es un lenguaje de programación originalmente desarrollado por Dennis M. Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

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.

La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990). La adopción de este estándar es muy amplia por lo que, si los programas creados lo siguen, el código es portable entre plataformas y/o arquitecturas.

El texto plano es el tipo de archivo más sencillo que hay, ya que en su contenido no hay otra cosa más que el texto que lo conforma.

ACTIVIDAD

Hacer una investigación muy somera acerca de los siguientes tipos de archivos:

- TXT
- Markdown
- HTML
- LaTeX
- CSV

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. El final del archivo se identifica habitualmente con un carácter especial definido como un marcador "end-of-file" (final de archivo), ubicado a continuación de la última línea de texto. Este tipo de contenedor incluye texto sin formato, pero tiene otras capacidades. Los archivos TXT sirven como almacenes de información a la vez que evitan las complicaciones propias de otros formatos de archivo. Los archivos afectados por la corrupción de datos son fácilmente recuperables y el usuario puede continuar su trabajo con la información restante. El inconveniente de utilizar archivos TXT proviene de su baja entropía, que provoca que los archivos TXT ocupen más espacio que otros archivos de texto.

Detalles técnicos sobre los archivos TXT

Los archivos TXT contienen un formato mínimo, pero cumplen con las definiciones de formato aceptadas por la terminal del sistema y los editores de texto simple. Los archivos TXT son universales debido a que cualquier procesador de texto puede leerlos. Los archivos TXT pueden utilizar Unicode para facilitar su lectura a usuarios de distintos idiomas. Los archivos de texto exclusivamente en código ASCII pueden

intercambiarse y leerse en Unix, Mac y Windows. El conjunto de caracteres más habitual es UTF-8, se diferencia del código ASCII en la marca de orden de bytes y es retrocompatible con ASCII.

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.

Como explica "John Gruber": <http://daringfireball.net/projects/markdown/>, uno de sus creadores, Markdown es realmente dos cosas: por un lado, el lenguaje; por otro, una herramienta de software que *convierte el lenguaje en HTML válido*.

Cómo es la sintaxis de Markdown

La sintaxis es muy sencilla y cuenta con varias opciones diferentes para algunos de sus elementos. Básicamente, se trata de añadir ciertos caracteres al inicio de la línea o antes y después de los elementos a los que vamos a aplicar el formato.

HTML: (Lenguaje de Marcas de Hipertexto, o Hypertext Markup Language en inglés) es el lenguaje usado para crear documentos web. HTML es un lenguaje de marcas, es decir, marca el texto y no debe ser confundido con un lenguaje de programación. Cuando se usa como extensión de archivo, ésta le informa a un explorador web como Firefox que el archivo es un documento web.

El hipertexto es texto mostrado en la computadora que es interactivo y tiene la capacidad de hacer referencia al texto de otro documento. La referencia a este texto se logra mediante el uso de vínculos. Al hacer clic en un vínculo serás transferido desde el documento original al documento vinculado.

Lenguaje de marcas

El lenguaje de marcas es un conjunto de etiquetas que describe el contenido y la apariencia de un documento cuando es mostrado en un explorador web. Estas etiquetas usan paréntesis para diferenciarse del texto. Por ejemplo, al usar la etiqueta *el explorador web es informado de que el formato del texto siguiente debe ser en cursivas*. A menudo las etiquetas se usan en pares, pero esto no ocurre siempre. El final de una etiqueta se indica con una barra inclinada, como en .

LaTeX: Los archivos con la extensión TEX son documentos de texto creados por un programa llamado LaTeX. LaTeX es un software de composición tipográfica que permite al usuario crear documentos con distintos diagramas, tablas y esquemas científicos de alta calidad. A diferencia de los procesadores de texto habituales, como Microsoft Word o Apple Pages, los documentos TEX se utilizan principalmente en distintas ramas científicas. Se utilizan tanto para crear documentos y artículos sencillos como para preparar libros y otros documentos técnicos para su publicación.

Detalles técnicos sobre los archivos TEX

Los documentos TEX de LaTeX se utilizan en informática, física, distintas ingenierías, lingüística experimental y otros diversos campos debido a la alta calidad de los documentos resultantes y al soporte de distintos tipos de contenido que ofrecen. Los documentos TEX soportan la inclusión de expresiones matemáticas complejas, gráficos científicos y otros símbolos propios de materias concretas. LaTeX se desarrolló en 1985 y actualmente es mantenido, actualizado y desarrollado por el proyecto LaTeX3.

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. Un fichero CSV es un archivo de texto que contiene una serie de valores separados por comas. Los valores pueden ser cualquier cosa, desde números de un presupuesto de una hoja de cálculo, hasta nombres y descripciones de una lista de clientes de un negocio.

¿Para qué sirven estos archivos?

Varios programas pueden crear y leer archivos CSV. Microsoft Excel utiliza extensivamente estos archivos, y Outlook puede exportar e importar archivos CSV para el transporte. Otras aplicaciones comunes que usan archivos CSV son OpenOffice, Microsoft Works, WordPerfect Office y NeoOffice.

Normalmente para importar o exportar de bases de datos de unas aplicaciones. Los programas de hojas de cálculo más habituales te dan la opción de grabar tu archivos en este formato, con la opción de “*guardar como*”.

EDITORES:

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

- | | |
|------------|----------------|
| ✓ Atom | ✓ Nano |
| ✓ Bluefish | ✓ Notepad++ |
| ✓ Brackets | ✓ Pico |
| ✓ Gedit | ✓ Sublime Text |
| ✓ Geany | ✓ Vim |
| ✓ Emacs | ✓ etc. |

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

- ◆ Resaltado de palabras clave
- ◆ Autocompletado
- ◆ Lista de elementos definidos
- ◆ Autosangrado
- ◆ Identificación de pares de paréntesis

- ◆ Integración de compilador
- ◆ Integración de control de versiones
- ◆ Integración de terminal
- ◆ Búsquedas avanzadas
- ◆ etc.

La siguiente actividad se realizó con Vim, solo se trató de abrir el tutor en la terminal e ir haciendo lo que te pedía, como se muestra a continuación:

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
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.
1
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.

86,0-1 9%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

=====
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 sea 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.

109,37-36 12%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
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.

157,0-1 18%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

=====
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.

177,59-58 21%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

=====
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,
6) El azúcar es dulce,
7) Y así eres tu.

=====
Lección 2.5: EL MANDATO DESHACER

224,7 27%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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 cambio; después #40 hace 64 segundos      245,62-61      30%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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?
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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 'inea').
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: MÁS CAMBIOS USANDO c      334,62-61      42%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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

-----
RESUMEN DE LA LECCIÓN 3      357,74-73      45%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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 / . Fijese que tanto el carácter /
   como el cursor aparecen en la última línea de la pantalla, lo mismo
   que el mandato : .
```



```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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. Anatria
Correo electrónico: eferna1@platea.pntic.mec.es

766,1 Final
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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 /. Fijese 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 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
princpio.

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

/errroor <INTRO>
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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 disperejos.

441,11 57%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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

~~~~~
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 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.

:s/laas/las/
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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

~~~~~
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.

:s/laas/las/ 458,1 59%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

main.pdf
MUAC.Ai Weiwei.docx'
orig_077ded2a6ebc5e2b546588b3edcdd12b(1).mp4'
orig_077ded2a6ebc5e2b546588b3edcdd12b(1).mp4.part'
orig_077ded2a6ebc5e2b546588b3edcdd12b(2).mp4'
orig_077ded2a6ebc5e2b546588b3edcdd12b(2).mp4.part'
orig_077ded2a6ebc5e2b546588b3edcdd12b.mp4
orig_077ded2a6ebc5e2b546588b3edcdd12b.mp4.part
PRÁCTICA_SFDP(1).pdf'
programa2019_tarde_mIERCOLES Y VIERNES(1).docx'
programa2019_tarde_mIERCOLES Y VIERNES.docx'
Sin título 1.docx'
TeachingFeeling2.5.apk
TeachingFeeling2.5.apk.part
vim-tutor.txt
y2mate.com - espaguetti_a_la_boloesa_bolognese_spaghetti_recipe_syMvcAsa2zY_360p
.mp4'
y2mate.com - recetas_faciles_con_salchicha_pQncewC2aig_360p.mp4'
y2mate.com - top_30_mejores_canciones_de_alan_walker_top_30_alan_walker_songs_V
ovtjP_dHk_360p.mp4'
y2mate.com - top_30_mejores_canciones_de_alan_walker_top_30_alan_walker_songs_V
ovtjP_dHk_360p.mp4.part'

Pulse INTRO o escriba una orden para continuar
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

~~~~~
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.

618,81-79 81%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

~~~~~
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.

607,0-1 81%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

~~~~~
Lección 6.2: EL MANDATO «APPEND» (añadir)

~~~~~

** Pulse a para insertar texto DESPUÉS del cursor. **

1. Mueva el cursor al final de la primera línea de abajo señalada con --->
pulsando $ en modo Normal.
2. Escriba una a (minúscula) para añadir texto DESPUÉS del carácter
que está sobre el cursor. (A mayúscula añade texto al final de la línea).
Nota: ¡Esto evita el pulsar i, el último carácter, el texto a insertar,
<ESC>, cursor a la derecha y, finalmente, x, sólo para añadir algo
al final de una línea!
3. Complete ahora la primera línea. Nótese que append es exactamente lo
mismo que modo Insert, excepto por el lugar donde se inserta el texto.
---> Esta línea le permitirá practicar el añadido de texto al final de una línea.
---> Esta línea le permitirá practicar el añadido de texto al final de una línea.

~~~~~
Lección 6.3: OTRA VERSIÓN DE «REPLACE» (reemplazar)
631,78-76 84%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

~~~~~
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.

~~~~~
664,0-1 87%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

~~~~~
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 búsque 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

:set hls is
683,20 89%
```



```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(Para el concepto de caja de una letra, véase la nota al final del fichero)

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

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

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

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

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

RESUMEN DE LA LECCIÓN 6

/ignore
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
help.txt For Vim version 8.0. Last change: 2017 Oct 28

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: Use ":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-T or 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
help.txt [Ayuda][R0]      1,1      Comienzo
vim-tutor.txt [+]         726,0-1    94%
help.txt [Solo lectura] 1343L, 51602C
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda

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>

731,0-1 96%
```

```
vanessa@Titan: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
error. cpo--.

4. Word motions word-motions

<S-Right> or [count] words forward. exclusive motion.
W
<C-Right> or [count] WORDS forward. exclusive motion.
W

e Forward to the end of word [count] inclusive.
Does not stop in an empty line.

E Forward to the end of WORD [count] inclusive.
Does not stop in an empty line.

<S-Left> or [count] words backward. exclusive motion.
b
<C-Left> or [count] WORDS backward. exclusive motion.
B

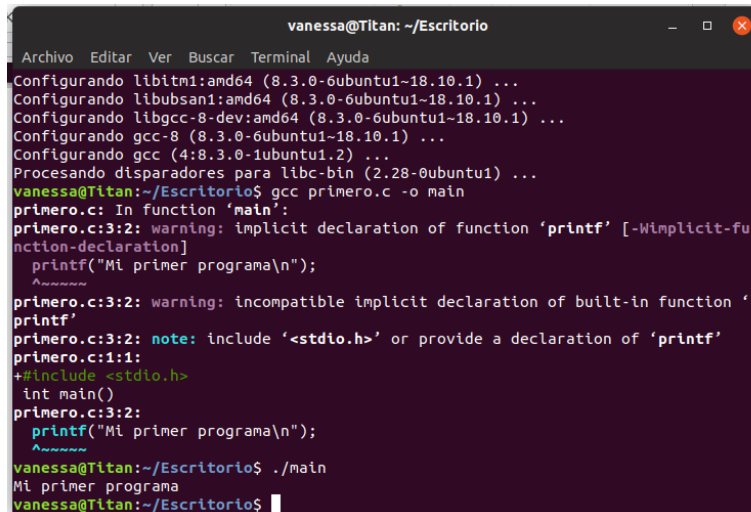
motion.txt [Ayuda][R0] 372,30-69 27%
vim-tutor.txt [+] 731,0-1 94%
"motion.txt" [Solo lectura] 1343L, 51602C
```

La siguiente actividad la realice con sublime text: un editor de código para lenguaje C. Iniciamos con un programa muy sencillo, el cual primero compilamos:

```
~/Escritorio/primer.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

primero.c
1 int main()
2 {
3     printf("Mi primer programa\n");
4
5     return 0;
6 }
7
8
```

y después lo corrimos:



```
vanessa@Titan: ~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
Configurando libitm1:amd64 (8.3.0-6ubuntu1-18.10.1) ...
Configurando libubsan1:amd64 (8.3.0-6ubuntu1-18.10.1) ...
Configurando libgcc-8-dev:amd64 (8.3.0-6ubuntu1-18.10.1) ...
Configurando gcc-8 (8.3.0-6ubuntu1-18.10.1) ...
Configurando gcc (4:8.3.0-1ubuntu1.2) ...
Procesando disparadores para libc-bin (2.28-0ubuntu1) ...
vanessa@Titan:~/Escritorio$ gcc primero.c -o main
primero.c: In function 'main':
primero.c:3:2: warning: implicit declaration of function 'printf' [-Wimplicit-fu
nction-declaration]
    printf("Mi primer programa\n");
    ^~~~~
primero.c:3:2: warning: incompatible implicit declaration of built-in function '
printf'
primero.c:3:2: note: include '<stdio.h>' or provide a declaration of 'printf'
primero.c:1:1:
+#include <stdio.h>
int main()
primero.c:3:2:
    printf("Mi primer programa\n");
    ^~~~~
vanessa@Titan:~/Escritorio$ ./main
Mi primer programa
vanessa@Titan:~/Escritorio$
```

y así fue como termino la práctica de ese día y por supuesto, nuestro pequeño programa.

CONCLUSIÓN:

Durante nuestra estancia en el laboratorio de computación estuvimos viendo los elementos básicos para un programa en lenguaje C y para eso fue necesario entender otros conceptos para poder compilar de forma correcta nuestro programa, lo que puedo decir es que, debes ser muy cuidadoso al momento de escribir en la terminal para poder compilar y correr el programa porque con un error que tengas todo puede ser fatal. En este lenguaje todo es muy importante cualquier punto, coma, espacio, etc. sin embargo, cuando logras tu objetivo es muy satisfactorio, o al menos, así lo fue para mí ya cuando lo entiendes y lo logras se siente muy bien.

REFERENCIAS DE LA INVESTIGACIÓN.

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

<https://www.genbeta.com/guia-de-inicio/que-es-markdown-para-que-sirve-y-como-usarlo>

https://techlandia.com/extension-archivo-html-hechos_525023/

<https://www.online-convert.com/es/formato-de-archivo/tex>

<https://lolap.wordpress.com/2015/01/14/que-es-un-csv-como-se-hace-y-para-que-sirve/>

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