



## 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:*

3

*No de Práctica(s):*

6

*Integrante(s):*

Velasco Gomez Noe Abimael

*No. de Equipo de  
cómputo empleado:*

13

*No. de Lista o Brigada:*

3989

*Semestre:*

2020-1

*Fecha de entrega:*

30/09/19

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

El objetivo de esta práctica es 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.

Y así se podremos tener un mayor conocimiento y entendimiento de los formatos de texto planos, para que sirven cada uno de ellos y mostrar lo simple, pero a la vez mostrar la utilidad de toso estos formatos, principalmente para saber hacer un desarrollo de algún programa pero en esta ocasión en lenguaje C.

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.

Algunos formatos:

## TXT

¿Qué es?

Es un texto plano en el cual se puede crear, abrir y editar utilizando una amplia variedad de programas de procesamiento de texto y de edición de textos desarrollados para sistemas Linux, ordenadores y plataformas Mac Microsoft basado en Windows. El contenido de estos .txt archivos de texto sin formato, es que se pueden guardar como .txt documentos en archivos de tamaño reducido.



```
Previsualización texto
7x8
[ 1]C[ ]*[ ]*[ 2]M[ ]*[ ]*[ ]*
[ 3]A[ ]V[ ]I[ ]O[ ]N[ ]*[ ]*
[ ]B[ ]*[ ]*[ ]T[ ]*[ ]*[ ]*
[ 4]A[ ]U[ ]T[ ]O[ 5]B[ ]U[ ]S
[ ]L[ ]*[ ]*[ ]*[ ]A[ ]*[ ]*
[ ]L[ ]*[ ]*[ ]*[ ]R[ ]*[ ]*
[ ]O[ ]*[ 6]C[ ]O[ ]C[ ]H[ ]E
[ ]*[ ]*[ ]*[ ]*[ ]O[ ]*[ ]*
Across
3. Medio de transporte aéreo|image/t_avion.gif
4. Transporte público por carretera|image/t_bus.gif
6. A mi me gustaría tener uno deportivo amarillo|image/t_coche
Down
1. Animal usado tradicionalmente para viajar|image/t_caballo.g
2. Vehículo a motor de dos ruedas|image/t_moto.gif
5. Medio de transporte marítimo|image/t_barco.gif
Tamaño fichero: 653 bytes
```

## MARKDOWN

¿Qué es? Markdown es una forma sencilla de agregar formato a textos web. Con formato nos referimos a itálicas, negritas, listas, y más, en texto plano, por ejemplo, el texto que nos encontramos en el visor de HTML de los editores de blogs, o en el Bloc de Notas de Windows, para hacer el concepto más asequible; Markdown fue creado originalmente por John Gruber, con ayuda de Aaron Swartz, con el propósito de crear un texto plano fácil de escribir y fácil de leer, y que pudiera convertirse de forma sencilla y válida a XHTML.

## Markdown

```
# This is Hello World in HTML

![A small picture of the Earth.]
(images/earth-small.png)

Hello World!

A.K.A:

*   Earth
*   Terra
*   Gaia
*   Globe
```

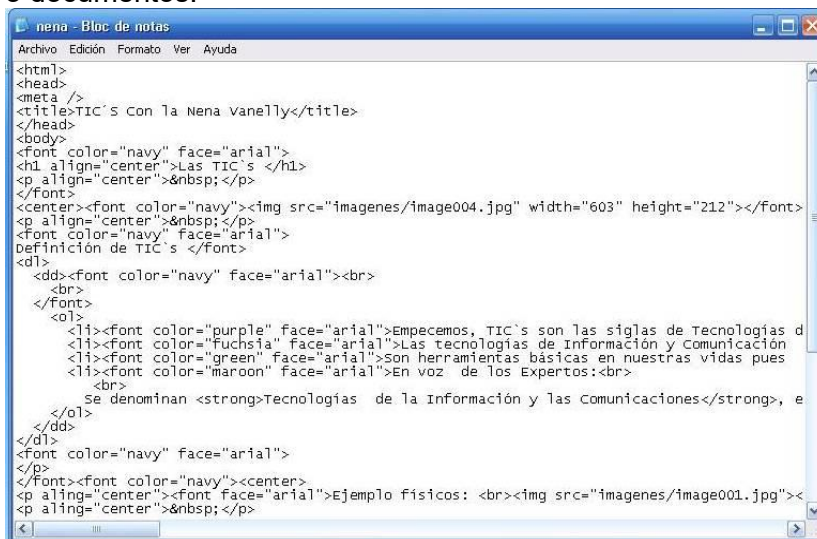
## HTML

¿Qué es?

HTML significa "Lenguaje de Marcado de Hypertexto" por sus siglas en inglés "HyperText Markup Language", es un lenguaje que pertenece a la familia de los "lenguajes de marcado" y es utilizado para la elaboración de páginas web.

Cabe destacar que HTML no es un lenguaje de programación ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de los lenguajes de programación, por lo que HTML genera únicamente páginas web estáticas, sin embargo, HTML se puede usar en conjunto con diversos lenguajes de programación para la creación de páginas web dinámicas.

Básicamente el lenguaje HTML sirve para describir la estructura básica de una página y organizar la forma en que se mostrará su contenido, además de que HTML permite incluir enlaces (links) hacia otras páginas o documentos.



```
nena - Bloc de notas
Archivo Edición Formato Ver Ayuda

<html>
<head>
<meta />
<title>TIC's Con la Nena Vanelly</title>
</head>
<body>
<font color="navy" face="arial">
<h1 align="center">Las TIC's </h1>
<p align="center">&nbsp;</p>
</font>
<center><font color="navy"></font>
<p align="center">&nbsp;</p>
<font color="navy" face="arial">
Definición de TIC's </font>
<dl>
<dd><font color="navy" face="arial"><br>
<br>
</font>
<ol>
<li><font color="purple" face="arial">Empecemos, TIC's son las siglas de Tecnologías d
<li><font color="fuchsia" face="arial">Las tecnologías de Información y Comunicación
<li><font color="green" face="arial">Son herramientas básicas en nuestras vidas pues
<li><font color="maroon" face="arial">En voz de los Expertos:<br>
<br>
Se denominan <strong>Tecnologías de la Información y las Comunicaciones</strong>, e
</ol>
</dd>
</dl>
</font>
<font color="navy" face="arial">
</p>
</font><font color="navy"><center>
<p align="center"><font face="arial">Ejemplo físicos: <br><
<p align="center">&nbsp;</p>
```

## LaTeX

¿Qué es? Es un sistema de composición de textos que está orientado especialmente a la creación de documentos científicos que contengan fórmulas matemáticas, cuadros y tablas. Además, también se pueden crear otros tipos de documentos, que pueden ser desde cartas sencillas hasta libros completos.

**Absolute Point Sizes**

size	standard classes (except <i>slides</i> ), beamer			AMS classes, <i>memoir</i>			slides
	[10pt]	[11pt]	[12pt]	[10pt]	[11pt]	[12pt]	
<code>\tiny</code>	5	6	6	6	7	8	13.82
<code>\scriptsize</code>	7	8	8	7	8	9	16.59
<code>\footnotesize</code>	8	9	10	8	9	10	16.59
<code>\small</code>	9	10	10.95	9	10	10.95	16.59
<code>\normalsize</code>	10	10.95	12	10	10.95	12	19.907
<code>\large</code>	12	12	14.4	10.95	12	14.4	23.89
<code>\Large</code>	14.4	14.4	17.28	12	14.4	17.28	28.66
<code>\LARGE</code>	17.28	17.28	20.74	14.4	17.28	20.74	34.4
<code>\huge</code>	20.74	20.74	24.88	17.28	20.74	24.88	41.28
<code>\Huge</code>	24.88	24.88	24.88	20.74	24.88	24.88	41.28

## CSV

¿Qué es?

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. Sirven 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”.

	A	B	C	D	E	F	G	H	I	J
1	Date,Open,High,Low,Close,Volume									
2	8-Dec-16	61.30	61.58	60.84	61.01	21043447				
3	7-Dec-16	60.01	61.38	59.80	61.37	30808969				
4	6-Dec-16	60.43	60.46	59.80	59.95	19907035				
5	5-Dec-16	59.70	60.58	59.56	60.22	23552658				
6	2-Dec-16	59.08	59.47	58.80	59.25	25515665				
7	1-Dec-16	60.11	60.15	58.94	59.20	34542121				
8	30-Nov-16	60.86	61.18	60.22	60.26	34655435				
9	29-Nov-16	60.65	61.41	60.52	61.09	22366721				
10	28-Nov-16	60.34	61.02	60.21	60.61	20732619				
11	25-Nov-16	60.30	60.53	60.13	60.53	8409616				
12	23-Nov-16	61.01	61.10	60.25	60.40	21848913				
13	22-Nov-16	60.98	61.26	60.80	61.12	23206700				
14	21-Nov-16	60.50	60.97	60.42	60.86	19652595				
15	18-Nov-16	60.78	61.14	60.30	60.35	27686311				

## Editor de texto de terminal:

### Actividad: Seguir el tutor de vim.

MINGW64:/c/Users/Invitado Isabel/Downloads

```
---> 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

**\*\* 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.

vim-tutor.txt[+] [unix] (16:04 30/09/2019)

MINGW64:/c/Users/Invitado Isabel/Downloads

#### Lección 1.4: EDICIÓN DE TEXTO - INSERCIÓN

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

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

---> Falta texto en esta . Buenas tardes profe, saqueme 10 porfavor :(

---> Falta algo de texto en esta línea. muchas gracias

5. Cuando se sienta cómodo insertando texto pase al resumen que esta más abajo.

#### RESUMEN DE LA LECCIÓN 1

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

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

Ahora continúe con la Lección 2.

#### Lección 2.1: MANDATOS PARA BORRAR

vim-tutor.txt[+] [unix] (16:04 30/09/2019)

-- INSERTAR --

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 pásalo bien que no pertenecen a esta frase.

#### Lección 2.2: MAS 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 (DESPUES del primer . ).
4. Escriba d\$ para borrar hasta el final de la línea.

---> Alguien ha escrito el final de esta línea dos veces|.

vim-tutor.txt[+] [unix] (16:04 30/09/2019)

#### 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 azucar 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. Para borrar desde el cursor hasta el final de una palabra pulse: dw
2. Para borrar desde el cursor hasta el final de una línea pulse: d\$

vim-tutor.txt[+] [unix] (16:04 30/09/2019)

1 cambio; antes #25 3 seconds ago

## Lección 3.1: EL MANDATO «PUT» (poner)

- \*\* Pulse p para poner lo último que ha borrado después del cursor. \*\*
1. Mueva el cursor al final de la lista de abajo.
  2. Escriba dd para borrar la línea y almacenarla en el buffer de Vim.
  3. Mueva el cursor a la línea que debe quedar por debajo de la línea a mover.
  4. Estando en mod Normal, pulse p para restituir la línea borrada.
  5. Repita los pasos 2 a 4 para poner todas las líneas en el orden correcto.
    - a) Las rosas son rojas,
    - b) Las violetas son azules,
    - c) La inteligencia se aprende,
    - d) ¿Puedes aprenderla tu?

## Lección 3.2: EL MANDATO «REPLACE» (reemplazar)

- \*\* Pulse r y un carácter para sustituir el carácter sobre el cursor. \*\*
1. Mueva el cursor a la primera línea de abajo señalada con --->.
  2. Mueva el cursor para situarlo bajo el primer error.
  3. Pulse r y el carácter que debe sustituir al erróneo.
  4. Repita los pasos 2 y 3 hasta que la primera línea esté corregida.
- > ¡cuando esta línea fue rescrita alguien pulsó algunas teclas equivocadas!  
 ---> ¡Cuando esta línea fue rescrita alguien pulsó algunas teclas equivocadas!

## Lección 3.3: EL MANDATO «CHANGE» (cambiar)

vim-tutor.txt[+] [unix] (16:04 30/09/2019)

-- INSERTAR --

## Lección 3.3: EL MANDATO «CHANGE» (cambiar)

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

1. Mueva el cursor a la primera línea de abajo señalada con --->.
  2. Sitúe el cursor en la u de lubrs.
  3. Escriba cw y corrija la palabra (en este caso, escriba 'línea').
  4. Pulse <ESC> y mueva el cursor al error siguiente (el primer carácter que deba cambiarse).
  5. Repita los pasos 3 y 4 hasta que la primera frase sea igual a la segunda.
- > Esta línea tiene unas pocas palabras que corregir usando el mandato change.  
 ---> Esta línea tiene unas pocas palabras que corregir usando el mandato change.

## Lección 3.4: 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\$.

## RESUMEN DE LA LECCION 3

1. Para sustituir texto que ha sido borrado, pulse p. Esto pone el texto

vim-tutor.txt[+] [unix] (16:04 30/09/2019)

-- INSERTAR --

## 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 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.

`<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 desaparejos.

## Lección 4.4: UNA FORMA DE CAMBIAR ERRORES

vim-tutor.txt[+] [unix] (16:04 30/09/2019)

/errroor <INTRO>

## 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. [D).

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

## 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.

---> 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.

## RESUMEN DE LA LECCION 4

vim-tutor.txt[+] [unix] (16:04 30/09/2019)



paréntesis, corchetes o llaves disparesos.

#### Lección 4.4: UNA FORMA DE CAMBIAR ERRORES

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

1. Mueva el cursor a la línea de abajo señalada con --->.
  2. Escriba :s/laas/las/ <INTRO> . Tenga en cuenta que este mandato cambia sólo la primera aparición en la línea de la expresión a cambiar.
- > las mejores épocas para ver las flores son las primaveras.
4. Para cambiar todas las apariciones de una expresión ente dos líneas escriba :#,#s/viejo/nuevo/g donde #,# son los números de las dos líneas. Escriba :%s/viejo/nuevo/g para hacer los cambios en todo el fichero.

#### RESUMEN DE LA LECCIÓN 4

1. Ctrl-g muestra la posición del cursor en el fichero y su estado. Mayu-G mueve el cursor al final del fichero. Un número de línea seguido de Mayu-G mueve el cursor a la línea con ese número.
2. Pulsando / seguido de una frase busca la frase hacia ADELANTE. Pulsando ? seguido de una frase busca la frase hacia ATRAS. Después de una búsqueda pulse n para encontrar la aparición siguiente en la misma dirección.
3. Pulsando % cuando el cursor esta sobre (,), [,], { o } localiza la pareja correspondiente.
4. Para cambiar viejo por nuevo en una línea pulse :s/viejo/nuevo  
Para cambiar todos los viejo por nuevo en una línea pulse :s/viejo/nuevo/g  
Para cambiar frases entre dos números de líneas pulse :#,#s/viejo/nuevo/g  
Para cambiar viejo por nuevo en todo el fichero pulse :%s/viejo/nuevo/g  
Para pedir confirmación en cada caso añada 'c' :%s/viejo/nuevo/gc

vim-tutor.txt[+] [unix] (16:04 30/09/2019)

versión del fichero.

#### RESUMEN DE LA LECCIÓN 5

1. **!:mandato** ejecuta un mandato externo.  
Algunos ejemplos útiles son:  
:!dir - muestra el contenido de un directorio.  
:!del NOMBRE\_DE\_FICHERO - borra el fichero NOMBRE\_DE\_FICHERO.
2. **:#,#w NOMBRE\_DE\_FICHERO** guarda desde las líneas # hasta la # en el fichero NOMBRE\_DE\_FICHERO.
3. **:r NOMBRE\_DE\_FICHERO** recupera el fichero del disco NOMBRE\_DE\_FICHERO y lo inserta en el fichero en curso a partir de la posición del cursor.

```
/ignore
```

---

#### RESUMEN DE LA LECCIÓN 6

1. Pulsando `o` abre una línea por DEBAJO del cursor y sitúa el cursor en la línea abierta en modo Insert.  
Pulsando una `O` mayúscula se abre una línea SOBRE la que está el cursor.
2. Pulse una `a` para insertar texto DESPUÉS del carácter sobre el cursor.  
Pulsando una `A` mayúscula añade automáticamente texto al final de la línea.
3. Pulsando una `R` mayúscula se entra en modo Replace hasta que, para salir, se pulse `<ESC>`.
4. Escribiendo `«:set xxx»` fija la opción «xxx»

---

#### 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

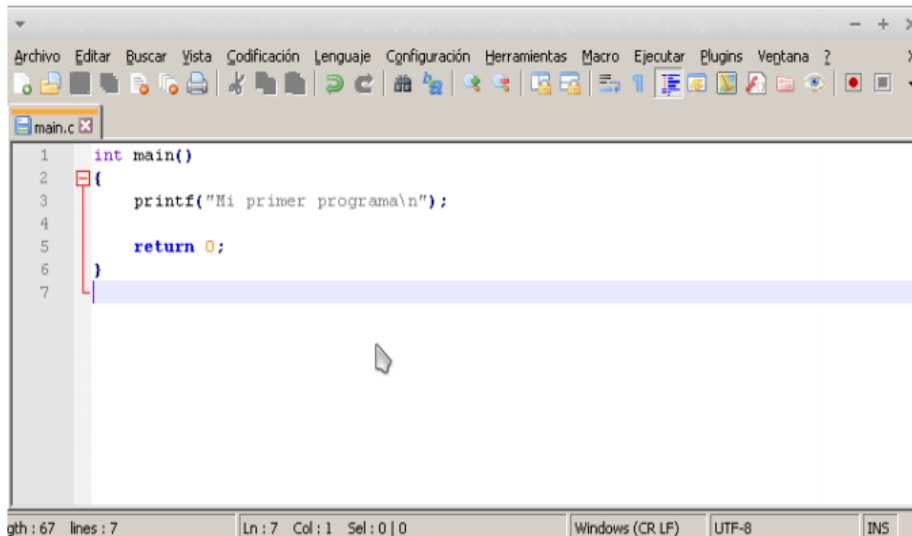
Vim-tutor.txt[+] [unix] (16:04 30/09/2019)



○ Escribe aquí para buscar



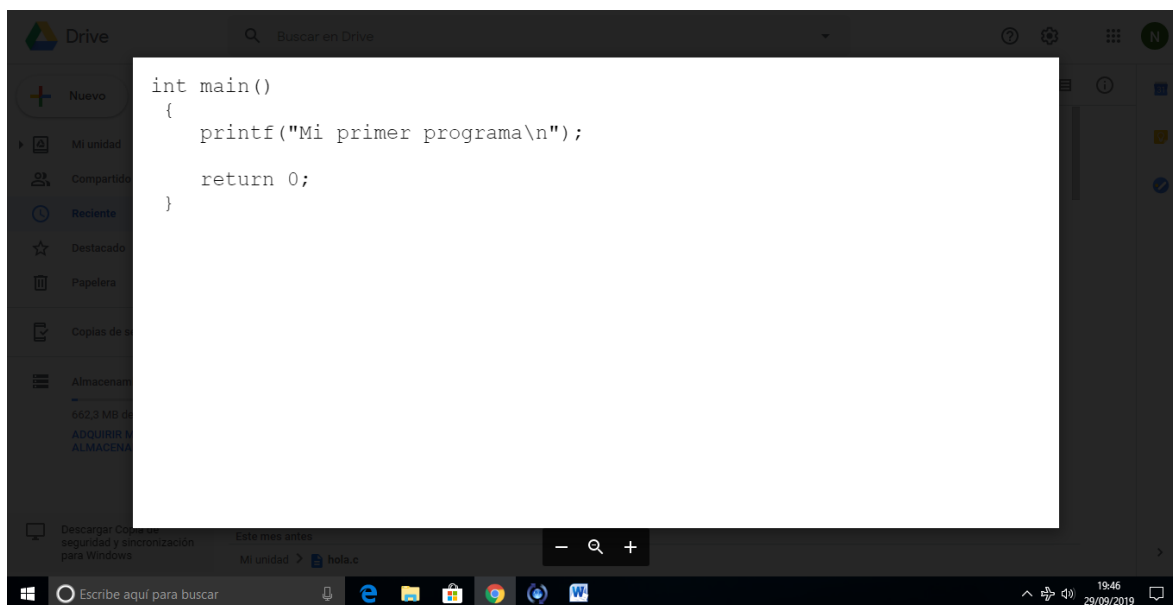
## Editor Grafico:



The screenshot shows a graphical code editor window with a menu bar (Archivo, Editar, Buscar, Vista, Codificación, Lenguaje, Configuración, Herramientas, Macro, Ejecutar, Plugins, Ventana, ?) and a toolbar. The editor displays a C program in a file named 'main.c'. The code is as follows:

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

The status bar at the bottom indicates 'gth : 67 lines : 7', 'Ln : 7 Col : 1 Sel : 0 | 0', 'Windows (CR LF)', 'UTF-8', and 'INS'.



The screenshot shows a Google Drive interface with a search bar and a sidebar on the left containing options like 'Nuevo', 'Mi unidad', 'Compartido', 'Reciente', 'Destacado', 'Papelera', 'Copias de seguridad', 'Almacenamiento', and 'Descargar Google Drive'. A code editor overlay is centered on the screen, displaying the same C program as the previous image:

```
int main()  
{  
    printf("Mi primer programa\n");  
    return 0;  
}
```

The Windows taskbar is visible at the bottom, showing the search bar, task view button, and several application icons. The system clock indicates 19:46 on 29/09/2019.

Puedo concluir que en esta práctica no hubo mayor complicación para entender a los formatos de texto plano y debo de mencionar que por lo menos la función del editor Sublime es muy útil y se maneja fácilmente para el desarrollo de algún programa por medio de lenguaje C, sin embargo si hubiese alguna complicación, aunque es muy mínima, sería el manejo de ciertos comandos tanto para el uso del lenguaje C, como del editor, específicamente hablando de cosas como el saber como es que se guarda correctamente el archivo a desear, pero como dije es un problema casi nulo. Y con el editor de texto plano Vim no tiene mayor complicación para usarse, pues sus comandos no son difíciles de aprenderse.