



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Alejandro Pimentel

*Asignatura:* Fundamentos de programación

*Grupo:* 3

*No de Práctica(s):* 6

*Integrante(s):* Colonia Montero Sonia

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* 8 (7455)

*Semestre:* 2020-1

*Fecha de entrega:* 30/09/2019

*Observaciones:* En las capturas que pones se pueden ver algunos errores, lo que no es malo, pero en la última compilación que se ve que haces ya no se alcanza a ver si esos errores se mantuvieron o no. Además también hace falta la ejecución, no solo debe compilar, también se debe ejecutar.

**CALIFICACIÓN:** 8

## Introducción

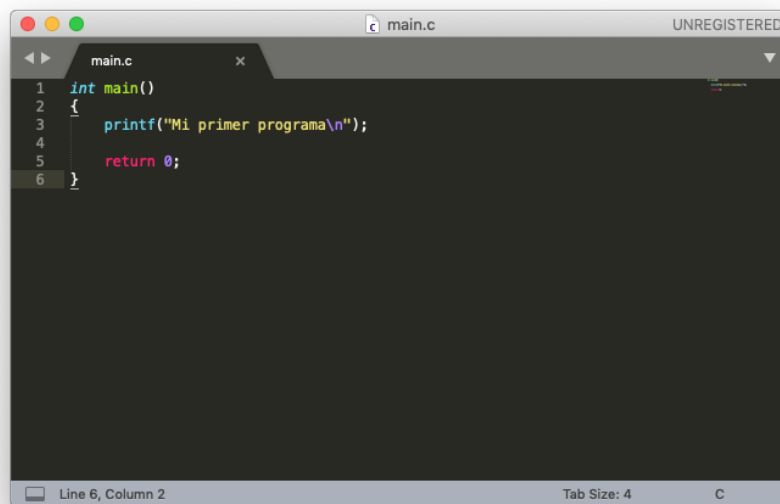
Existen varios ambientes y herramientas que sirven para el desarrollo y la ejecución de programas en Lenguaje C. Por ejemplo, existe una gran cantidad de tipos de archivos, los cuales se mencionarán a lo largo del presente documento. Uno de ellos es el texto plano, el cual es el tipo de archivo más sencillo que hay, ya que en su contenido no hay otra cosa más que el texto que lo conforma. Por otro lado, un editor de texto es un programa capaz de editar texto plano como lo son Atom, Bluefish, Brackets, Gedit, Geany, Emacs, Nano, Notepad++, Pico, Sublime Text, Vim, entre otros. Por lo general, los editores difieren en su modo de uso y en las características que ofrecen las cuales pueden ser 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, entre otras.

## 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.

## Desarrollo

Después de investigar algunos tipos de archivos; se usó Sublime Text para dar un comando sencillo y con el objetivo principal de conocerlo.



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

Line 6, Column 2      Tab Size: 4      C

Posteriormente, dicho archivo creado se corrió desde la terminal como se muestra a continuación.

Singapur40:~ fp03alu08\$ cd Desktop **%cambiar a la carpeta donde se guardó el archivo**

Singapur40:Desktop fp03alu08\$ gcc main.c -o main **%correr programa**

```
Escritorio — -bash — 80x24
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
main.c:3:9: warning: missing terminating '"' character [-Winvalid-pp-token]
      printf("Mi primer programa\n");
      ^
main.c:3:9: error: expected expression
main.c:6:2: error: expected '}'
}
^
main.c:2:1: note: to match this '{'
{
^
2 warnings and 2 errors generated.
[Singapur40:Desktop fp03alu08$ gcc main.c -o main
main.c:3:2: warning: implicitly declaring library function 'printf' with type
      'int (const char *, ...)' [-Wimplicit-function-declaration]
      printf("Mi primer programa\n");
      ^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
main.c:3:9: warning: missing terminating '"' character [-Winvalid-pp-token]
      printf("Mi primer programa\n");
      ^
main.c:3:9: error: expected expression
main.c:6:2: error: expected '}'
}
^
main.c:2:1: note: to match this '{'
{
^
2 warnings and 2 errors generated.
[Singapur40:Desktop fp03alu08$ gcc main.c -o main
main.c:3:2: warning: implicitly declaring library function 'printf' with type
      'int (const char *, ...)' [-Wimplicit-function-declaration]
      printf("Mi primer programa\n");
```

Singapur40:Desktop fp03alu08\$ ./main  
Mi primer programa

Por último, se usó Vim y algunos de sus mandatos.

## Resultados

- **Txt:** La extensión TXT representa "textfile". Este tipo de contenedor incluye texto sin formato y sirve como almacén de información a la vez que evitan las complicaciones propias de otros formatos de archivo.
- **Markdown:** Facilita la aplicación de formato a un texto empleando una serie de caracteres de una forma especial. Se puede emplear para cualquier tipo de texto aunque en general se usa para HTML. Al utilizar la sintaxis de Markdown, se pueden producir archivos legibles como texto plano y que a la vez están listos para ser formados en otras plataformas. Muchos generadores de bitácoras y de sitios estáticos, así como sitios como GitHub aceptan Markdown y traducen estos archivos a HTML para su visualización en la web.

- **Html:** Es el estándar para la creación de sitios web. Los navegadores web comprenden este lenguaje y pueden interpretar su codificación en diferentes textos, colores, formatos e hiperenlaces, así como insertar imágenes y audio mediante la incorporación de URL.
- **LaTeX:** Es un sistema de preparación de documentos. Con él se puede preparar manuscritos, artículos de revista, cartas, tesis, presentaciones y cualquier tipo de documento para imprimir en papel o mostrar en pantalla.
- **Csv:** El CSV significa "valores separados por comas" debido a que los datos de estos archivos CSV están divididos por comas en conjuntos particulares de información. Filas base de datos independientes están representados por cada línea de texto que se almacena en un archivo CSV. Estas filas de bases de datos se implementan con uno o más campos de datos, y estos se dividen por comas.

A continuación se muestran algunos de los mandatos de los que dispone Vim.

h	Izquierda
k	Arriba
l	Derecha
j	Abajo
<ESC>	Asegurarse que está en modo normal o regresar a él.
:q! <INTRO>	Salir del editor sin guardar cambios
:wq <INTRO>	Salir del editor guardando los cambios
vim <i>tutor</i> <INTRO>	Para entrar al editor en el <i>fichero</i> deseado
x	Permite borrar carácter
i	Permite insertar caracteres
dw	Borrar de donde se ponga el cursor al final de la palabra
d\$	Borrar el final de la línea donde se ponga el cursor
dd	Borrar para borrar líneas enteras
[número] d objeto O d [número] objeto	número - cuántas veces se ha de ejecutar el mandato. d - mandato para borrar.                      c-mandato de cambiar objeto - sobre lo que el mandato va a operar.
	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.
u	Deshacer los últimos mandatos
U	Devolver la línea a su estado original
CTRL-R	Deshacer lo deshecho.
p	Poner el texto borrado después del cursor
r + nuevo carácter	Sustituye al carácter sobre el que está el cursor
cw + lo que sea	Cambiar parte de la palabra o toda ella
CTRL-g	Mostrar su situación en el fichero y su estado
MAYU-G	Moverse a una determinada línea en el fichero
/ + frase	Buscar la frase deseada
	n                      Repetir la búsqueda
	Mayu-N              Buscar la misma frase en dirección opuesta
	? en lugar de /      Buscar una frase en dirección opuesta
%	Encontrar paréntesis correspondiente al paréntesis sobre el que este el cursor.
:s/viejo/nuevo	Para cambiar palabra “viejo” por palabra “nuevo” en una línea

:s/viejo/nuevo/g	Para cambiar todas las palabras “viejo” por palabra “nuevo” en una línea
:#,#s/viejo/nuevo/g	Para cambiar palabra “viejo” por palabra “nuevo” entre dos números de línea
:%s/viejo/nuevo/g	Para cambiar palabra “viejo” por palabra “nuevo” en todo el fichero
:s/viejo/nuevo/gc	Para pedir confirmación en cada caso
:! + mandato	Ejecutar dicho mandato externo (ej. ls)
:W NOMBRE_DE_FICHERO	Guardar cambios hechos en un fichero
:#,# NOMBRE_DEL_FICHERO	Guardar desde la línea # hasta la # en un fichero
:r NOMBRE_DEL_FICHERO	Insertar el contenido de un fichero
o	Abrir línea debajo del cursor y pone cursor en dicha línea en modo insertar
O	Abre línea encima del cursor
a	Insertar texto después del cursor
A	Añadir texto al final de la línea
R	Sustituir más de un carácter
<<set xxx>>	Fija la opción xxx (ej. :set ic para ignorar la caja de cierta letra)
<AYUDA>	Activar sistema de ayuda
<F1>	
:help <INTRO>	
:q <INTRO>	Cerrar la ventana de ayuda

## Conclusión

Existe una gran cantidad de tipos de archivos y cada uno de ellos tiene características y funciones diferentes que permiten realizar cierto tipo de documentos o trabajos. Es importante conocer por lo menos los más importantes de ellos ya que así se puede usar el tipo de archivo que mejor se ajuste a lo que se quiere crear. Por otro lado, los editores de texto son importantes para la modificación, creación e incluso almacenamiento de archivos. Mediante el presente documento se logró conocer los comandos básicos de C así como usar conocer y usar herramientas para el desarrollo y ejecución de programas en Lenguaje C.

## Referencias

- ✓ *TXT, Raw text file (.txt)*. Recuperado el 29 de septiembre de 2019, de Online-convert: <https://www.online-convert.com/es/formato-de-archivo/txt>
- ✓ Simpkin, Sarah (2015) *Introducción a Markdown*. Recuperado el 29 de septiembre de 2019, de The programming Historian: <https://programminghistorian.org/es/lecciones/introduccion-a-markdown>
- ✓ *TXT, Raw text file (.txt)*. Recuperado el 29 de septiembre de 2019, de Online-convert: <https://www.online-convert.com/es/formato-de-archivo/html>
- ✓ (2014) *Guía rápida LaTeX*. Recuperado el 29 de septiembre de 2019, de LaTeX Fácil: <http://nokyotsu.com/latex/guia.html>
- ✓ Corel Corporation *.CSV Extensión del archivo*. Recuperado el 29 de septiembre de 2019, de ReviverSoft: <https://www.reviversoft.com/es/file-extensions/csv>