



## 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 programación

*Grupo:*

3

*No de Práctica(s):*

6

*Integrante(s):*

Nava Corona Nadia Erandeni

*No. de Equipo de cómputo  
empleado:*

*No. de Lista o Brigada:*

6948

*Semestre:*

2020-1

*Fecha de entrega:*

Lunes 30 de septiembre

*Observaciones:*

Tus conclusiones son pobres. Pero más importante que eso, te faltan evidencias (capturas) de la ejecución correcta de tu programa.

**CALIFICACIÓN:** 8

# "Entorno de C"

## Introducción.

C es un lenguaje de programación de propósito general. Se trata de un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel, ya que 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.

## Objetivo.

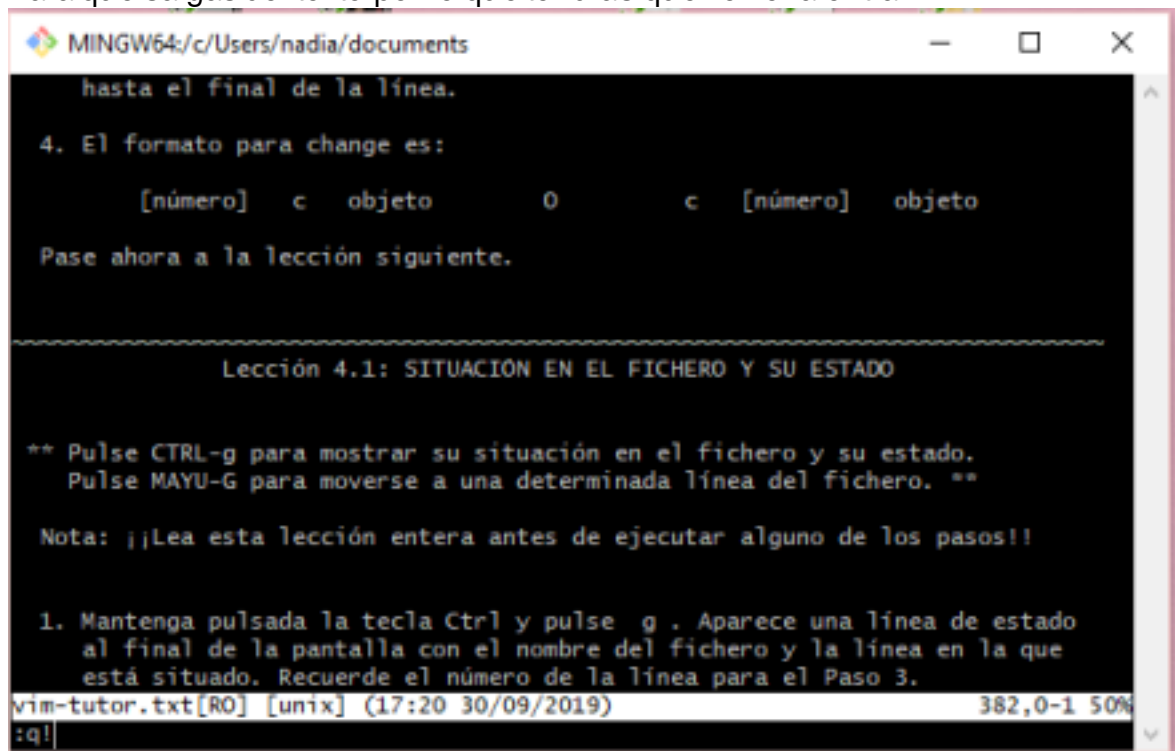
Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

**Actividad 1.** Hacer una investigación muy somera acerca de los siguientes tipos de archivos:

- **Txt:** archivo de texto plano como lo puede ser el block de notas.
- **Markdown:** es un lenguaje de marcado sencillo que sirve para agregar formato, vínculos e imágenes con facilidad al texto simple.
- **HTML:** siglas en inglés de HyperText Markup Language ('lenguaje de marcas de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.
- **LaTeX:** 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.
- **Csv:** Los archivos CSV) 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 y las filas por saltos de línea. El formato CSV es muy sencillo y no indica un juego de caracteres concreto, ni cómo van situados los bytes, ni el formato para el salto de línea. Estos puntos deben indicarse muchas veces al abrir el archivo, por ejemplo, con una hoja de cálculo.

## Actividad 2. Seguir el tutor de vim

1. La primera lección que se tiene es el movimiento de cursor. Con la letra "J" se mueve hacia abajo, con la letra "K" se mueve hacia arriba, con la letra "H" se mueve hacia la izquierda y con la tecla "L" se mueve hacia la derecha.
2. La siguiente lección es salirte del texto, para ello debes asegurarte que estás en modo normal tecleando "esc" y después escribes ":q!" y das enter. Esto hará que salgas del texto por lo que tendrás que volver a entrar.



```
MINGW64:/c:/Users/nadia/documents
hasta el final de la línea.

4. El formato para change es:

      [número] c objeto      0      c [número] objeto

Pase ahora a la lección siguiente.

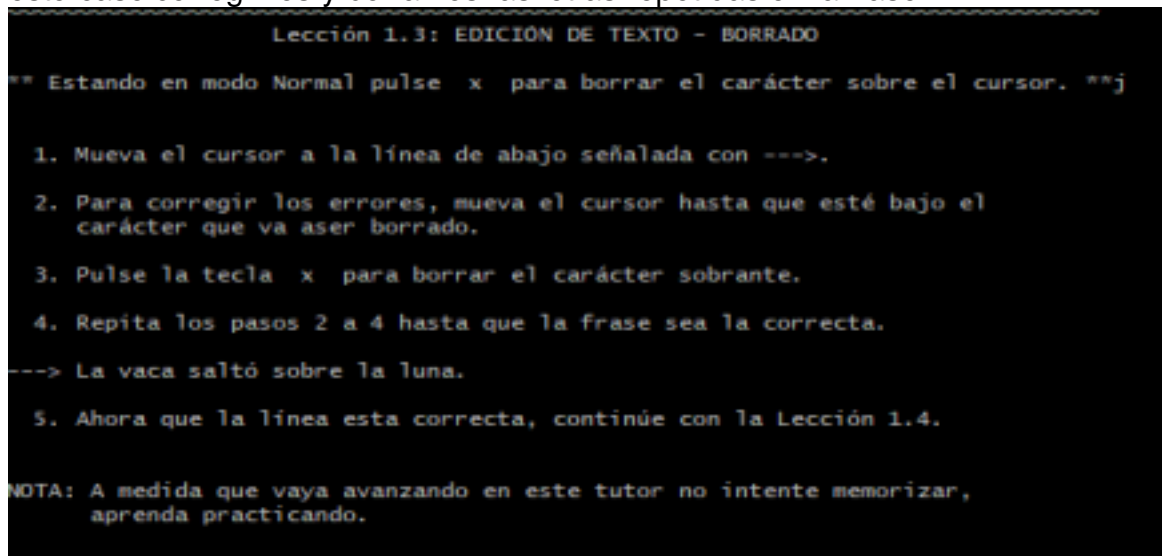
-----
                        Lección 4.1: SITUACIÓN EN EL FICHERO Y SU ESTADO

** Pulse CTRL-g para mostrar su situación en el fichero y su estado.
   Pulse MAYU-G para moverse a una determinada línea del fichero. **

Nota: ¡¡Lea esta lección entera antes de ejecutar alguno de los pasos!!

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.
vim-tutor.txt[RO] [unix] (17:20 30/09/2019) 382,0-1 50%
:q!
```

3. En esta lección, aprendemos a borrar un carácter solamente usando la tecla "x", debes colocar el cursor justo debajo de la letra que desees borrar. En este caso corregimos y borramos las letras repetidas en la frase.



```
MINGW64:/c:/Users/nadia/documents
-----
                        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.

vim-tutor.txt[RO] [unix] (17:20 30/09/2019) 382,0-1 50%
:q!
```

4. Para agregar texto debes dar click en la tecla “i” (recuerda que debes asegurarte de estar en modo normal con “esc”) Y una vez que hayas acabado vuelves a teclear “esc” para regresar a modo normal.

```
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 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.
```

5. A diferencia del paso anterior “x” para borrar, debes emplear “dw” para borrar una palabra completa.

```
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.
```

6. Para borrar varias palabras debes emplear “\$d”, esta borra las palabras escritas después de un punto.

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

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

7. El comando “dd” sirve para borrar toda una línea y si colocas “2dd” se borra la línea donde colocaste el cursor y la línea de abajo.

## 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.

8. El comando “u” sirve para deshacer el comando que has hecho, en este caso con “x” corriges un carácter y con “U” se deshace todos los comandos hechos en toda la línea

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.

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.

9. El comando “p” es utilizado para poner lo último que ha sido borrado. Usamos “dd” para borrar toda una línea (esta se queda almacenada), movemos el cursor a donde queremos colocar la línea y tecleamos la letra “p”. Así hasta corregir la lista

```
a) Las rosas son rojas,  
b) Las violetas son azules,  
c) La inteligencia se aprende,  
d) ¿Puedes aprenderla tu?
```

10. El comando “r” sirve para remplazar un carácter por otro. Debes colocar el cursor debajo del carácter a reemplazar, teclear la letra “r” y el carácter que desees poner a cambio de ese.

```
---> ¡Cuando esta línea fue escrita alguien pulso algunas teclas equi|vocadas!  
---> ¡Cuando esta línea fue escrita alguien pulsó algunas teclas equivocadas!
```

11. El comando change “cw” sirve para cambiar una palabra. Debes colocar el cursos donde desees cambiar la palabra, usar cw y escribir la palabra que desees.

```
---> Esta línea tiene unas pocas palabras que corregir usando el mandato change.  
---> Esta línea tiene unas pocas palabras que corregir usando el mandato change.
```

12. Para borrar todo lo que resta en una línea, debes colocar el cursor debajo de la primera palabra a borrar, usas el comando “c\$” y se borra lo que resta de la línea, además de que te permite editar el texto.

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

13. El comando “ctrl g” sirve para saber el estado del fichero (saber dónde está) y la información aparecerá al final de la pantalla.

```
Vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019) 401,67-65 51%  
"vim-tutor.txt" [Modificado][Sólo lectura] 766 lines --52%--
```

Y el comando “G” sirve para irte hasta al final del texto.

```
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  
  
-----  
vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019) 766,1 Final
```

14. Para buscar una palabra debes irte al final del texto y escribir `/lapalabraquequieresbuscar` y automáticamente te llevará hacia allá

```
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.

vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019)
/errroor|
```

15. El comando `%` te ayuda a identificar un par de corchetes, esto para que sea más fácil la distinción de estos

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

16. El comando `:s/viejo/nuevo/g` sirven para sustituir las palabras que se escribieron después de las diagonales

```
--> Las mejores épocas para ver las flores son las primaveras.
```

17. Para ejecutar un mandato externo debes emplear el comando `:!`  y el comando que desees, en este caso fue `ls`

```
vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019)
:!!ls
[No se ha escrito nada al disco desde el último cambio]
desktop.ini Instrucciones.txt main.C vim-tutor.txt
```

18. Para guardar ficheros se debe usar el comando `:w nombredelfichero` por ejemplo y para comprobar que se ha hecho usa el comando `!dir`

```
Lección 5.3: UN MANDATO DE ESCRITURA SELECTIVO
vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019)
:w TEST
desktop.ini Instrucciones.txt main.C vim-tutor.txt

Pulse INTRO o escriba una orden para continuar
[No se ha escrito nada al disco desde el último cambio]
desktop.ini Instrucciones.txt main.C TEST vim-tutor.txt
```

19. Para guardar solamente una parte de fichero debes utilizar el comando `:#, #w nombredetufichero`, por ejemplo `:70, 72 w TEST`

20. El comando `:r nombredetufichero` sirve para volver a poner el texto de tu fichero.



21. Si tecleas la letra o minúscula agrega una línea por debajo del cursor si tecleas una O mayúscula agrega una línea por arriba del cursor

```

** 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.

vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019) 1388,82-80 90%
-- INSERTAR --
```

22. El mandato APPEND que se escribe "a" sirve para añadir o insertar 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.
```

23. El mando REPLACE usado con la letra R mayúscula sirve para remplazar caracteres.

```

--> 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.
```

24. El comando `:set xxx` fija la opción xxx

```

MINGW64/c:/Users/nadia/documents

** 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

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

vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019) 690,16 44%
:set hls is
```



## 25. Comando :help

```
MINGW64: C:/Users/nadia/documents
help.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,      k
                   "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
help.txt[Ayuda][-][RO] [unix] (08:41 17/08/2019)      1,1 Comienzo
mandato «:help» mandato. Pruebe estos:
vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019)      732,43-40 95%
"help.txt" [Sólo lectura][UNIX] 238L, 8894C
```

## Comando :help w

```
MINGW64: c:/Users/nadia/documents
error. cpo--.

=====
4. Word motions                                word-motions
<S-Right>      or      [count] words forward.  exclusive motion.  <S-Right> |w
<C-Right>      or      [count] WORDS forward.  exclusive motion.  <C-Right> |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.

motion.txt[Ayuda][-][RO] [unix] (07:10 24/06/2019)      370,30-69 27%
mandato «:help» mandato. Pruebe estos:
vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019)      732,43-40 95%
"motion.txt" [Sólo lectura][UNIX] 1327L, 50875C
```

## Comando :help c\_<T

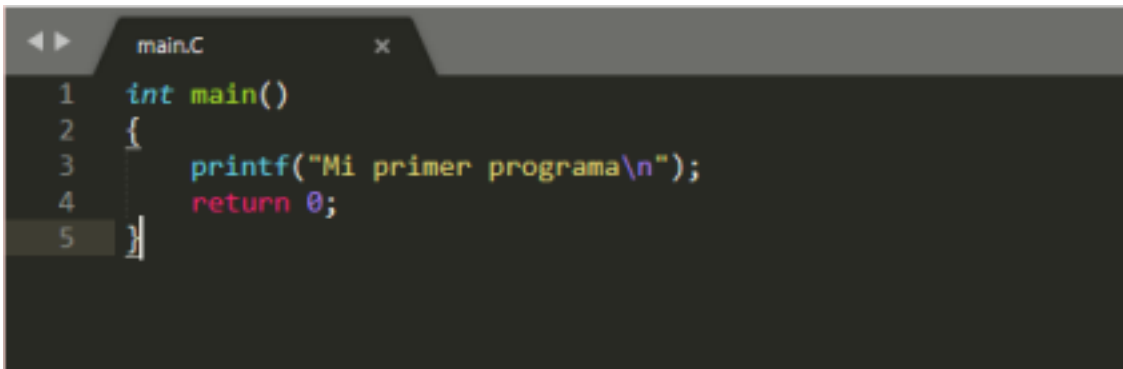
```
MINGW64/c/Users/nadia/documents
When showing file names, directories are highlighted (see
'highlight' option). Names where 'suffixes' matches are moved
to the end.
The 'wildoptions' option can be set to "tagfile" to list the
file of matching tags.
c_CTRL-I c_wildchar |c_<Tab>
'wildchar' option
A match is done on the pattern in front of the cursor. The
match (if there are several, the first match) is inserted
in place of the pattern. (Note: does not work inside a
macro, because <Tab> or <Esc> are mostly used as 'wildchar',
and these have a special meaning in some macros.) When typed
again and there were multiple matches, the next
match is inserted. After the last match, the first is used
again (wrap around).
The behavior can be changed with the 'wildmode' option.
c_<S-Tab>
<S-Tab> Like 'wildchar' or <Tab>, but begin with the last match and
then go to the previous match.
cmdline.txt[Ayuda][-][RO] [unix] (07:10 24/06/2019) 397,30-65 33%
mandato «:help» mandato. Pruebe estos:
vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019) 732,43-40 95%
"cmdline.txt" [S lo lectura][UNIX] 1202L, 48412C
```

Comando :help insert-index

```
MINGW64/c/Users/nadia/documents
For an overview of built-in functions see functions.
For a list of Vim variables see vim-variable.
For a complete listing of all help items see help-tags.
=====
1. Insert mode |insert-index
tag char action in Insert mode
-----
i_CTRL-@ CTRL-@ insert previously inserted text and stop
insert
i_CTRL-A CTRL-A insert previously inserted text
CTRL-B not used i_CTRL-B-gone
i_CTRL-C CTRL-C quit insert mode, without checking for
abbreviation, unless 'insertmode' set.
i_CTRL-D CTRL-D delete one shiftwidth of indent in the current
line
i_CTRL-E CTRL-E insert the character which is below the cursor
CTRL-F not used (but by default it's in 'cinkeys' to
index.txt[Ayuda][-][RO] [unix] (14:25 03/07/2019) 32,21-57 1%
mandato «:help» mandato. Pruebe estos:
vim-tutor.txt[+][RO] [unix] (17:20 30/09/2019) 732,43-40 95%
"index.txt" [S lo lectura][UNIX] 1728L, 79491C
```

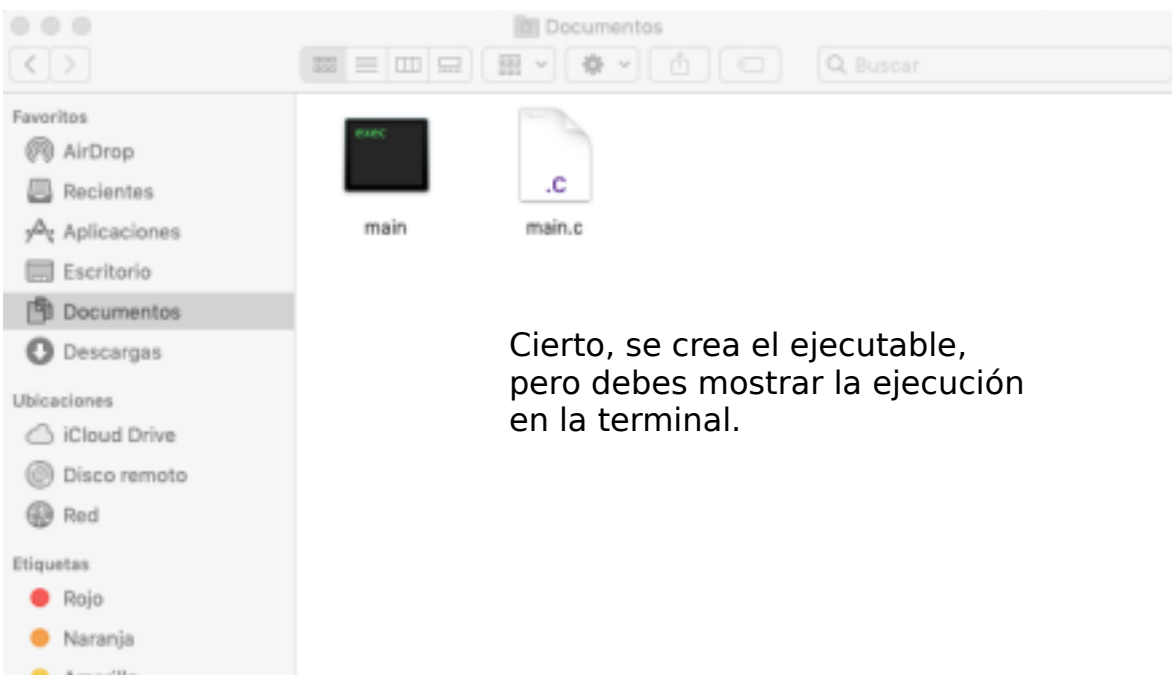
### Actividad 3.

Creemos un programa con lenguaje en C



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

Para que este se ejecute en la terminal, debemos compilarlo y correrlo, si tu código está correcto aparecerá de esta manera



## Conclusiones.

Este tutorial de vim nos enseña a modificar un texto plano, desde aprender a movernos en él, hasta añadir, borrar, buscar palabras.