



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: 135

No de Práctica(s): 6

Integrante(s): Páez Matínez Karen

*No. de Equipo de
cómputo empleado:* 2

No. de Lista o Brigada: 3781

Semestre: 1

Fecha de entrega: 30/09/2019

Observaciones: Excelente. Haz lo posible por dejar la portada en una sola hoja.

CALIFICACIÓN:

10

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.

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

- **txt:**

Un archivo de texto simple, texto sencillo o texto sin formato (también llamado texto llano o texto simple; en inglés «plain text»), es un archivo informático que contiene únicamente texto formado solo por caracteres que son legibles por humanos, careciendo de cualquier tipo de formato tipográfico. Estos archivos están compuestos de bytes que representan caracteres ordinarios como letras, números y signos de puntuación (incluyendo espacios en blanco), también incluye algunos pocos caracteres de control como tabulaciones, saltos de línea y retornos de carro. Estos caracteres se pueden codificar de distintos modos. El sistema de codificación ASCII viene a ser la base primordial y no necesita de un identificador explícito en la comunicación digital.

- **Markdown:**

Markdown fue desarrollado en 2004 por John Gruber, y se refiere tanto a (1) una manera de formar archivos de texto, como a (2) una utilidad del lenguaje de programación Perl para convertir archivos Markdown en HTML. En esta lección nos centraremos en la primera acepción y aprenderemos a escribir archivos utilizando la sintaxis de Markdown.

- **Html:**

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. Es un estándar a cargo del Word Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la Word Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

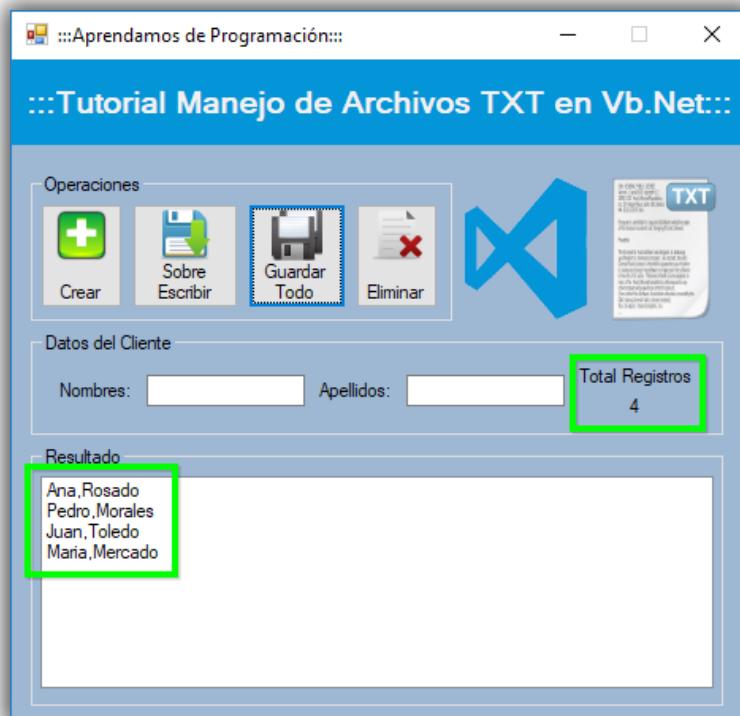
- **LaTeX:**

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.

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, 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:**

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.



ACTIVIDAD#1 "Vim":

1. Primero descargamos el programa después empezamos seguir los pasos del tutorial de Vim:

```
** Escriba dd para borrar una línea entera. **

Debido a la frecuencia con que se borran líneas enteras, los desarrolladores de Vim desarrollaron qué sería más fácil el escribir simplemente dos d's 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. Mueva ahora a la cuarta línea.
4. Escriba 2dd (recuerde número-mandato-objeto) para borrar las dos líneas.

1) Las rosas son rojas,
2) El cielo es azul,
3) El mar es dulce,
4) Y así eres tú.

Lección 2.5: EL MANDATO DESHACER

vim-tutor.txt(+) [smr] (12:54 29/09/2013) 224,7 27
```

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

1. Mueva el cursor a la línea de abajo señalada con ---- y síntalo 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 corrige 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 deshacer).

---- Corrija los errores de esta línea y vuélvala a ponerlos con deshacer.
8. Estos mandatos son muy útiles. Ahora pase al resumen de la Lección 2.

RESUMEN DE LA LECCIÓN 2
vim-tutor.txt(+) [smr] (12:54 29/09/2013) 245,8 30%
```

```
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!
```

- Después ubicar la carpeta en donde descargamos el tutorial de Vim después de esto vamos a ingresar con vi y con el nombre del archivo.

```
ACERQUEDO MINIGUI - (master)
$ cd Downloads/
ACERQUEDO MINIGUI - ~/Downloads (master)
$ vi vim-tutor.txt
```

```
!! 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 siente con confianza, ejecute los
pasos 1 a 3 para salir y volver a entrar al editor. Después muévase el
cursor hasta La Lección 1.3.
vim-tutor.txt [unix] (12:54 29/09/2019) 54,0-1 6h
[un]
```

- Después seguimos siguiendo en tutorial con los diferentes pasos:

The terminal window displays four different Vim help pages:

- Lesson 4.3: EDICIÓN DE TEXTO - BORRADO**: Shows steps for deleting text in Normal mode, including moving the cursor to a line, using the 'd' command, and navigating back to the start of the line.
- Lección 4.3: EDICIÓN DE TEXTO - INSERCION**: Shows steps for inserting text in Normal mode, including moving the cursor to the first character of a line, inserting text, and navigating back to the start of the line.
- Lección 5.1: MÁS PANTALLAS PARA BORRAR**: Shows steps for deleting multiple lines, including moving the cursor to the end of a line, using the 'dd' command, and navigating back to the start of the line.
- Lección 5.2: MÁS PANTALLAS PARA BORRAR**: Shows steps for deleting text from the current line to the end of the file, including moving the cursor to the end of the line, using the 'dd' command, and navigating back to the start of the line.

```

--> qwrite .d para borrar una linea entera. -->
Borrado a la Frase anterior con que se borran lineas enteras. Los desplazamientos de Vim desplazan con que seria mas facil el escribir simplemente dos dos en una fila para borrar una linea.

2. Pausa el cursor a la segunda linea de la linea de abajo.
3. Escritura dd para borrar la linea.
4. Muestra ahora a la cuarta linea.
5. Escritura 2dd (recuerda numero mandato-objeto) para borrar las dos lineas.

--> Los errores son errores.
--> El resto no errores.
--> El error es un delfito.
7. Y asi eres tu.

L00758 3.3: EL BORRADO DESPLAZAR
vim -u /etc/vimrc & vim /etc/vimrc
2dd,2dd

```

```

--> pulsa x para deshacer los ultimos mandatos.
--> o para deshacer una linea entera.

1. Muestra el cursor a la linea de abajo señalada con ---- y arrastra hacia el primer error.
2. Pulsa x para borrar el primer caracter erroneo.
3. Pulsa ahora x para deshacer el ultimo mandato ejecutado.
4. Ahora corrija todos los errores de la linea usando el mandato x.
5. Pulsa ahora x (siguiente linea para deshacer la linea a su estado original).
6. Pulsa errores veces para volver para deshacer lo hecho por x y los siguientes errores.
7. Ahora pulsa CTRl-R (buscando palabra la linea cito, y en la K) para posar errores para volver a ejecutar los mandatos (deshacer lo deshecho).

--> Corrige los errores del anterior linea y vuelve a ponerlos con destino.
8. Los mandatos son muy difftes. Ahora pasa al resumen de la Lección 3.

RESUMEN DE LA LECCION 3
vim -u /etc/vimrc & vim /etc/vimrc
2dd,2dd

```

```

LECCION 3.2: EL BORRADO -REPLACE- (reemplazar)
vim -u /etc/vimrc & vim /etc/vimrc
x,x

--> pulsa r y un carácter para sustituir el carácter sobre el cursor. -->

1. Muestra el cursor a la primera linea de abajo señalada con ----.
2. Muestra el cursor para situarlo bajo el primer error.
3. Pulsa r y el carácter que debe sustituir al anterior.
4. Repite los pasos 3 y 4 hasta que la primera linea este corregida.

--> cuando esta linea fue escrita algunas pulsadas teclas espaciadoras
--> cuando esta linea fue escrita algunas pulsadas teclas espaciadoras

--> Para cambiar parte de una palabra o toda ella escrita. de : -->

1. Muestra el cursor a la primera linea de abajo señalada con ----.
2. Muestra el cursor en la u de haber.
3. Escribre cu y corrige la palabra (en este caso, escribe 'linea').
4. Pulsa dd y muestra el cursor al error siguiente (el primer carácter que debe corregir).
5. Repite los pasos 3 y 4 hasta que la primera linea sea igual a la segunda.

--> Esta linea tiene unas pocas palabras que corrige usando el mandato change.
--> Esta linea tiene unas pocas palabras que corrige usando el mandato change.

LECCION 3.4: MAS COMANDOS USANDO :
vim -u /etc/vimrc & vim /etc/vimrc
:dd,dd

```

```

--> Para cambiar parte de una palabra o toda ella escrita. de : -->

1. Muestra el cursor a la primera linea de abajo señalada con ----.
2. Muestra el cursor en la u de haber.
3. Escribre cu y corrige la palabra (en este caso, escribe 'linea').
4. Pulsa dd y muestra el cursor al error siguiente (el primer carácter que debe corregir).
5. Repite los pasos 3 y 4 hasta que la primera linea sea igual a la segunda.

--> Esta linea tiene unas pocas palabras que corrige usando el mandato change.
--> Esta linea tiene unas pocas palabras que corrige usando el mandato change.

LECCION 3.4: MAS COMANDOS USANDO :
vim -u /etc/vimrc & vim /etc/vimrc
:dd,dd

```

4. En esta parte vemos como esta el fichero y su situacion como lo dice el tutorial de Vim.

```

    ** Para cambiar parte de una palabra o toda ella escrita, se :
    1. Mueve el cursor a la primera línea de abajo señalada con >-->.
    2. Títtée el cursor en la u de Tutor.
    3. Escribir, por si corrige la palabra (en este caso, escribe "linea").
    4. Pulse Bksp y muove el cursor al error siguiente (el primer carácter que debía cambiarse).
    5. Escribir los pasos 3 y 4 hasta que la primera frase sea igual a la segunda.

    Esta frase tiene unas pocas palabras que corregir usando el mandato change.
    Esta frase tiene unas pocas palabras que corregir usando el mandato change.

    Lección 4.1: MANDADOS USANDO <INTRO>

    --> El mandato change se utiliza con los mismos datos que delete.
    <introduction> [units] (22:54 29/09/2019) 437,27-438,402

```

Para Texturas y estadios posteriores, te recomienda el libro:

«Texturas y estadios para el teclado» de Luisa Gómez
Editorial Bell & Howell & Asociados S.A.

Es un libro libre para llegar a saber todo lo que deseas hacer con Vi. La sexta edición incluye también informaciones sobre VIM.

Este material ha sido escrito por Michael G. Moore y Robert A. Morris. Traducción de textos y frases diferentes vienesas suministradas por Charles Smith, Colaborador Oficial de la Comunidad de Vim.

E-mail: vim-tutor@vim.org.es

Traducido para VIM por Juan Muñoz.

Editorial F. Adolfo
Correo electrónico: fadolfo@fadi.es, adolfo@es

vim-tutor.txt [units] (22:54 29/09/2019) 438,4-439,1

5. Aquí se da cuenta de uno de como funciona para buscar las palabras:

```

    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 error <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.
    errorrr no es la forma de deletrear error; errorrr es un error.

    vim-tutor.txt [units] (22:54 29/09/2019) 437,22-549

```

```

    Lección 4.2: EL MANDATO «SEARCH» (buscar)

    ** Escriba / seguido de una frase para buscar la frase. **
    3. 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 error <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.
    errorrr no es la forma de deletrear error; errorrr es un error.

    vim-tutor.txt [units] (22:54 29/09/2019) 437,22-549

```

6. Aquí seguimos lo que sigue después del funcionamiento de palabras:

```

    Lección 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS

    ** Pulse N 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 N .
    3. El cursor debería situarse en el paréntesis (, corchete [ o llave { correspondiente.
    4. Pulse N 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 desparejados.

    vim-tutor.txt [units] (22:54 29/09/2019) 447,11-570

```

```

    Lección 4.3: BÚSQUEDA PARA COMPROBAR PARENTESIS

    ** Pulse N 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 N .
    3. El cursor debería situarse en el paréntesis (, corchete [ o llave { correspondiente.
    4. Pulse N 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 desparejados.

    vim-tutor.txt [units] (22:54 29/09/2019) 447,11-570

```

7. Aquí vemos como sale de título de una forma de cambiar errores:

```
Lección 4.4: UNA FORMA DE CAMBIAR ERRORES

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

. Mueva el cursor a la linea de abajo señalada con --->.

. Escriba :s/laas/las/ <INTRO> . Tenga en cuenta que este mandato cambia
sólo la primera aparición en la linea de la expresión a cambiar.

> Laas mejores épocas para ver laas flores son laas primaveras.

. Para cambiar todas las apariciones de una expresión entre 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.

-tutor.txt [unix] (12:54 29/09/2019) 464,36-35 60%
laas/las
```

```
st_arm.exe
Lope+Derr+Fuenteovejuna.pdf
0180413_183315.3gp
0180418_184825.3gp
0180420_184653.3gp
0180427_185537.3gp
retirada (1).docx'
retirada.docx
utor.txt
NATO
11-dj-7.0 (1).exe'
11-dj-7.0.exe
11-dj-8-5-3573.msi
app Image 2018-10-17 at 9.41.18 PM.jpeg'
app Image 2019-01-13 at 7.06.50 PM (1).jpeg'
(1).docx'
docx
compat.diagcab
halifa - See You Again ft. Charlie Puth [Official Video] Furious 7 Soundtr
'
r-132.docx

INTRO o escriba una orden para continuar
```

8. Aquí también con ayuda del tutorial la lección 7 nos dice cómo escribir un comando con ayuda:

```

Rápido verás todos los comandos que tiene Vim.
1. Insertar la opción "help" (presione la tecla de fin de línea para terminar)
:help F1
2. Muestra todos los "tags" de Vim y pregunta si deseas cambiar entre más información. La salida es:
:help tags
3. Muestra las opciones "abbreviations" y "abbreviations"
:help abbreviations
4. Salir de Vim
:q
5. Salir de Vim sin guardar cambios
:q!

```

RETURN TO THE LESSON 6

En esta parte utilizamos la tecla F1

9. O también podemos ocupar el “help”

```

For vim version 8.0. Last change: 2019 Jun 21.
VM - main help file

Move around: use the cursor keys, or "h" to go left, "j" to go down, "k" to go up, "l" to go right.
Close this window: use "q". Careful, all changes are lost.
Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-].
Get specific help: It is possible to go directly to whatever you want help on. Type :help <subject>. Prepend something to specify the context: :help-context <subject>

```

10. Por ultimo somo seguimos los ultimos pasos y ya:

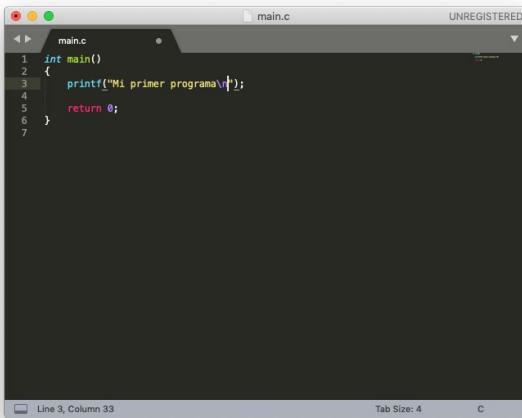
W

c_<T

Insert-index

“Notepad++”

1. Aquí en la clase empezamos a realizar el ejemplo de cómo empezarlo a realizar en sublime text. Empezamos a seguir los pasos de la práctica 6,y lo guardamos en un archivo en este caso lo guarde en una de documentos.

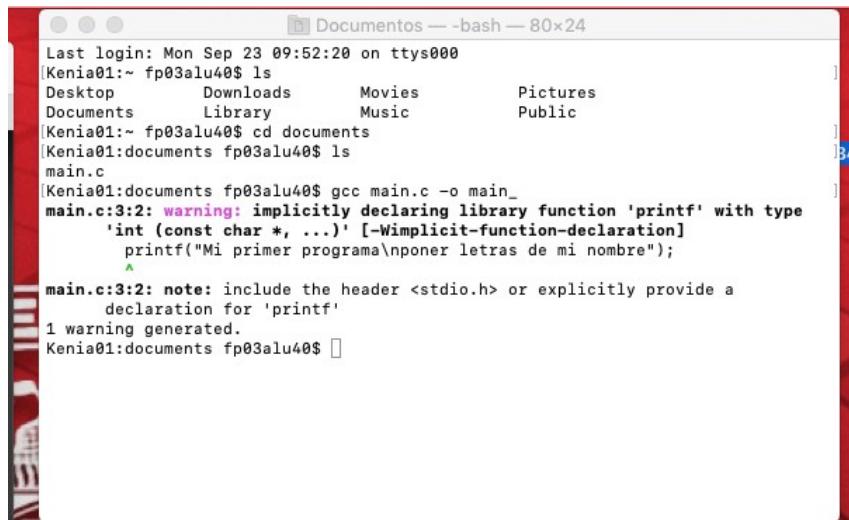


A screenshot of the Notepad++ code editor. The window title is "main.c". The code in the editor is:

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

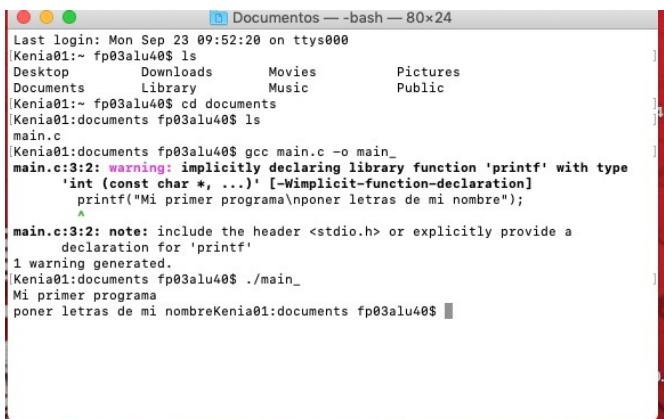
The status bar at the bottom shows "Line 3, Column 33" and "Tab Size: 4".

2. Después de haberlo guardado en documentos abrimos el terminador, y escogemos la de documentos y de ahí vamos siguiendo los pasos de la práctica #6.

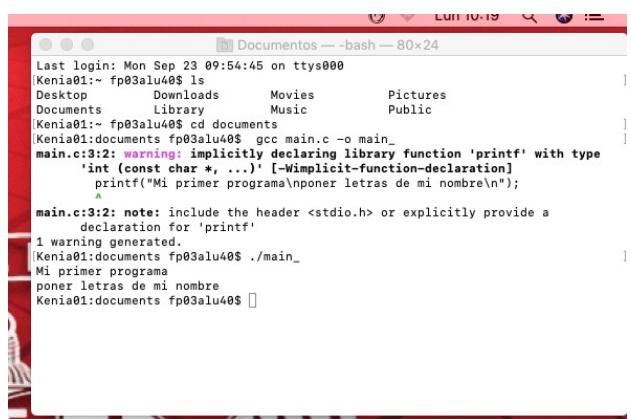


A screenshot of a terminal window titled "Documentos — bash — 80x24". The terminal output is as follows:

```
Last login: Mon Sep 23 09:52:20 on ttys000
[Kenia01:~ fp03alu40$ ls
Desktop      Downloads      Movies      Pictures
Documents    Library       Music       Public
[Kenia01:~ fp03alu40$ cd documents
[Kenia01:documents fp03alu40$ ls
main.c
[Kenia01:documents fp03alu40$ 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\nponer letras de mi nombre");
    ^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
[Kenia01:documents fp03alu40$ ]
```



```
Last login: Mon Sep 23 09:52:20 on ttys000
Kenia01:~ fp03alu40$ ls
Desktop      Downloads      Movies      Pictures
Documents    Library       Music       Public
Kenia01:~ fp03alu40$ cd documents
Kenia01:documents fp03alu40$ ls
main.c
Kenia01:documents fp03alu40$ 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\nponer letras de mi nombre");
^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
Kenia01:documents fp03alu40$ ./main_
Mi primer programa
poner letras de mi nombre
Kenia01:documents fp03alu40$
```



```
Last login: Mon Sep 23 09:54:45 on ttys000
Kenia01:~ fp03alu40$ ls
Desktop      Downloads      Movies      Pictures
Documents    Library       Music       Public
Kenia01:~ fp03alu40$ cd documents
Kenia01:documents fp03alu40$ 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\nponer letras de mi nombre");
^
main.c:3:2: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
Kenia01:documents fp03alu40$ ./main_
Mi primer programa
poner letras de mi nombre
Kenia01:documents fp03alu40$
```

CONCLUSION:

En esta práctica se me dificultó un poco al tratar de bajar los programas, pero al realizarlos creo que fue más sencillo porque me gusto seguir las especificaciones del profesor fue algo interesante y divertido aprender cosas nuevas como que el `\` significa un texto plano y que existe varios tipos de este y en diferentes archivos de texto plano como los que el profesor explicó en la práctica #6, en lo particular siento que me salió bien el trabajo aun que me costó trabajo.