



## 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):* SOTELO LEYVA CARLOS

*No. de Equipo de  
cómputo empleado:* 35

*No. de Lista o Brigada:* 47

*Semestre:* 2020-1

*Fecha de entrega:* 01/10/2019

*Observaciones:* Práctica correcta.  
Pero la entrega se hizo a destiempo

**CALIFICACIÓN:** 8

Lenguaje de programación En informática, cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora. Es complicado definir qué es y qué no es un lenguaje de programación. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Normalmente es la computadora la que realiza la traducción.

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 sistema, aunque también se utiliza para crear aplicaciones.

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.

#### 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:

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.

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

#### ¿Qué significa TXT?

Documentos de texto plano guardados en formato TXT se pueden 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 ASCII sin formato es que se pueden guardar como .txt documentos en archivos de tamaño reducido.

#### ¿Qué es un Markdown?

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.

¿Qué significa HTML?

HTML significa Hypertext Markup Language, y estos archivos HTML se implementan en su mayoría en forma de páginas estáticas de sitios web. HTML se puede utilizar para integrar ciertos atributos de formato y las especificaciones de diseño en el contenido de estas páginas web. Hay una gran cantidad de aplicaciones que se pueden utilizar para crear y editar estos archivos HTML, y muchos de los navegadores web se pueden usar para abrir y ver el contenido de archivos en él .HTML formato.

¿Qué es LaTeX?

La extensión de archivo LATEX se utiliza para un sistema de documentos de preparación destinada a la composición tipográfica de alta calidad. Por lo general se prefiere para medianas y grandes documentos científicos o técnicos, sino también utilizado para la publicación de cualquier forma. Esto permite a los autores que utilizan este sistema para dar información básica, como el texto del artículo, título, nombre del autor, la fecha y el título del documento.

¿Qué es 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.

EDITORES:

A continuación vimos algunos editores primordiales como los fueron:

- Atom
- Bluefish
- Brackets
- Gedit
- Geany
- Emacs
- Nano
- Notepad++
- Pico
- Sublime Text

- Vim
- etc.

En el cual el profesor nos dijo que el Notepad++ nos seria de mayor ayuda

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

NANO:

En este apartado de la práctica el profesor nos dice que es uno de editor de texto de terminal, ya que es muy fácil y útil de usar ya que se parece mucho al hacer commits de git como lo hemos venido haciendo.

Y vimos unas nuevas ordines que Podemos darle a git:

- git config --global core.editor nano
- git config --global core.editor nano
- git config --global core.editor vi

Vi:

Otro editor de termina, aunque también tiene versiones gráficas.

Este es uno de los editores más populares, con más características, y que los expertos aseguran que ofrecen mayor ganancia de velocidad para editar textos.

La razón es que este editor es bastante diferente a lo que están acostumbrados. Se requiere más tiempo para aprenderlo a usar y para dominarlo.

El tutor de vim nos dice varias lecciones de mandatos para darle a la pc, las primeras lecciones nos dan una entrada sobre como se empezarán a hacer las demás, poco a poco lo que vimos primero nos va sirviendo para las lecciones próximas, lo esencial nos dicen como se deben de manejar las teclas H, J, L, ya que estas nos sirven para mover la pantalla.

En las siguientes lecciones nos dice como entra y salir de vim, con la Tecla “esc” y la palabra Vim tutor <INTRO>.

La proxima lección nos sirve para editar el texto, estando normal la pc, presionamos la letra “I” para insertar el texto, solamente Debemos decir donde queramos insertar el texto nuevo y ya se queda editado.

La lección 2 nos dice que para borrar desde los cursores hasta el final de una palabra o frase solo pongamos: dw. Lo siguiente nos dice que para borrar desde los cursores hasta el final de la linea de la cual estemos escribiendo pulsamos: d\$. Para borrar una linea enter pulsamos las letras: dd.

El formato en modo normal es:

[número] mandato objeto o mandato [número] objeto donde:

Número: es cuantas veces se tiene que ejecutar.

Mandato: es lo que se tiene que hacer.

Objeto: es lo que el mandato tiene que operar.

También nos dice esto que para deshacer acciones previas es presionando: u (minúscula). Para deshacer todos los cambios de una línea es pulsando U (mayúscula). Para deshacer lo deshecho pulsando: CTRL-R.

La lección 3 nos dice que para sustituir texto que ya ha sido eliminado, pulsemos p, esto hace que el texto borrado o eliminado aparezca después del cursor, también nos dice que para sustituir el carácter bajo el cursor, pulsamos r y luego el carácter que sera sustituido del original. Otro dato es que el mando change le permite cambiar el objeto especificado desde la posición hasta el final del objeto, pulsando cw para cambiar desde el cursor al final de la palabra.

La lección 4 nos habla sobre cosas más complejas como lo son: ctrl-g muestra la posición del cursor en el fichero y su estado, Mayu-g mueve el cursor al final del fichero, un número en línea seguido de Mayu-g mueve al cursor a la línea de ese número. Nos dice que pulsando / seguido de una frase busca la frase hacia Adelante, pulsando ? seguido de una frase busca la frase hacia atrás, después de cada búsqueda pulsamos n para encontrar la aparición siguientes.

Otra cosa más de esta práctica es que pulsando % cuando el cursor esta sobre (,), [,], { o } así localizando la pareja que le corresponda. Para cambiar Viejo por nuevo en una linea: s/ Viejo/ nuevo. Para cambiar todos los viejos por nuevos en una línea pulse : s/ Viejo/ nuevo g. Para cambiar de frases entre 2 números de lineas pulsamos: #, #s / Viejo/ nuevo/ g. Para cambiar Viejo por nuevo en todo el fichero pulsamos: %s / Viejo/ nuevo/ g. Para pedir confirmaciones en cada caso añadimos “c”: %s/ Viejo / nuevo/ gc.

La lección 5 nos dice que pongamos : !mandato para que ejecute un mandato externo. Algunos de los ejemplos útiles son:

- :! Dir – muestra el contenido de un directorio.
- :! Del NOMBRE \_ DE \_ FICHERO – borra el fichero NOMBRE \_ DE FICHERO.

: # , #w NOMBRE \_ DE \_ FICHERO guarda desde las líneas # hasta la # en el fichero NOMBRE \_ DE \_ FICHERO.

: 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.

La lección 6 nos dice que pulsando o nos abre una línea por debajo del cursor y sitúan el cursor en la línea abierta en modo Insert. Pulsamos una O mayúscula se abre una línea sobre la que está en el cursor.

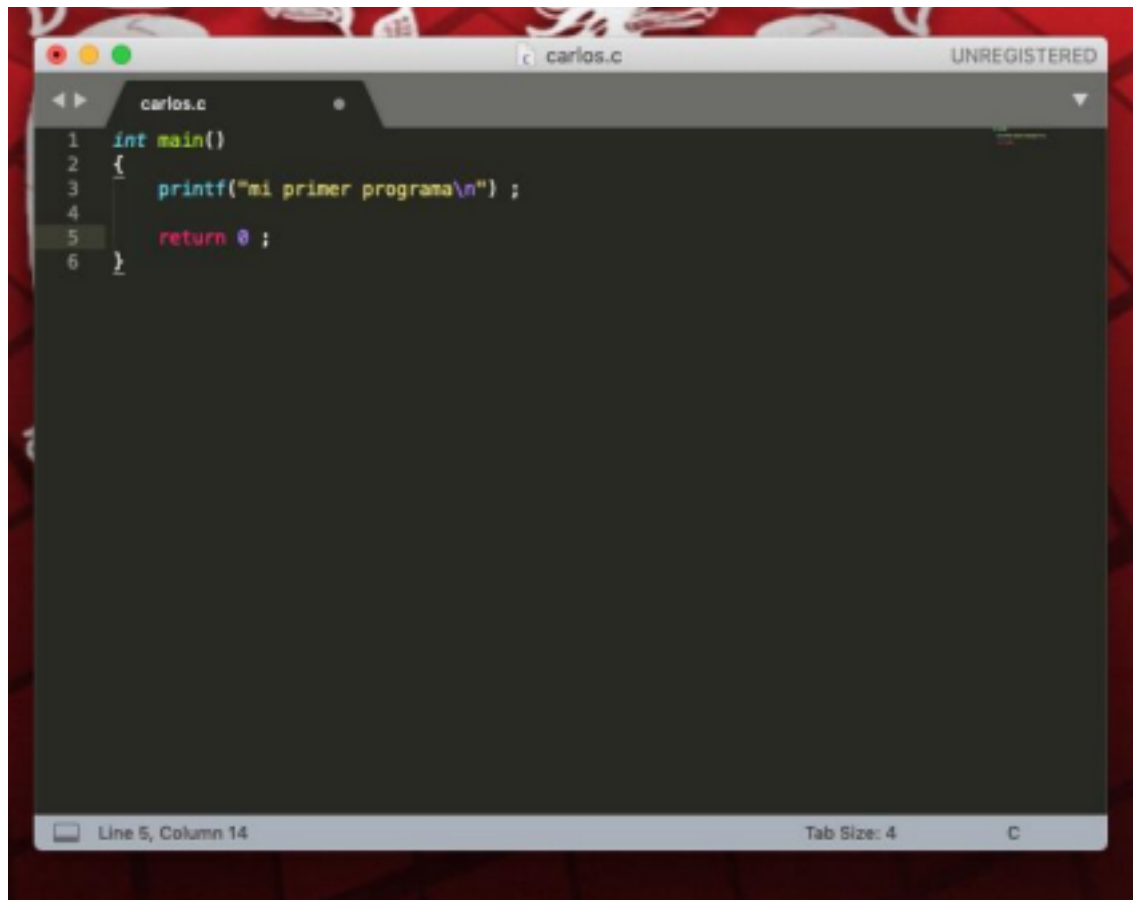
Pulsamos una a para insertar texto después del carácter sobre el cursor.

Pulsando una A mayúscula añade automáticamente que texto al final de la línea.

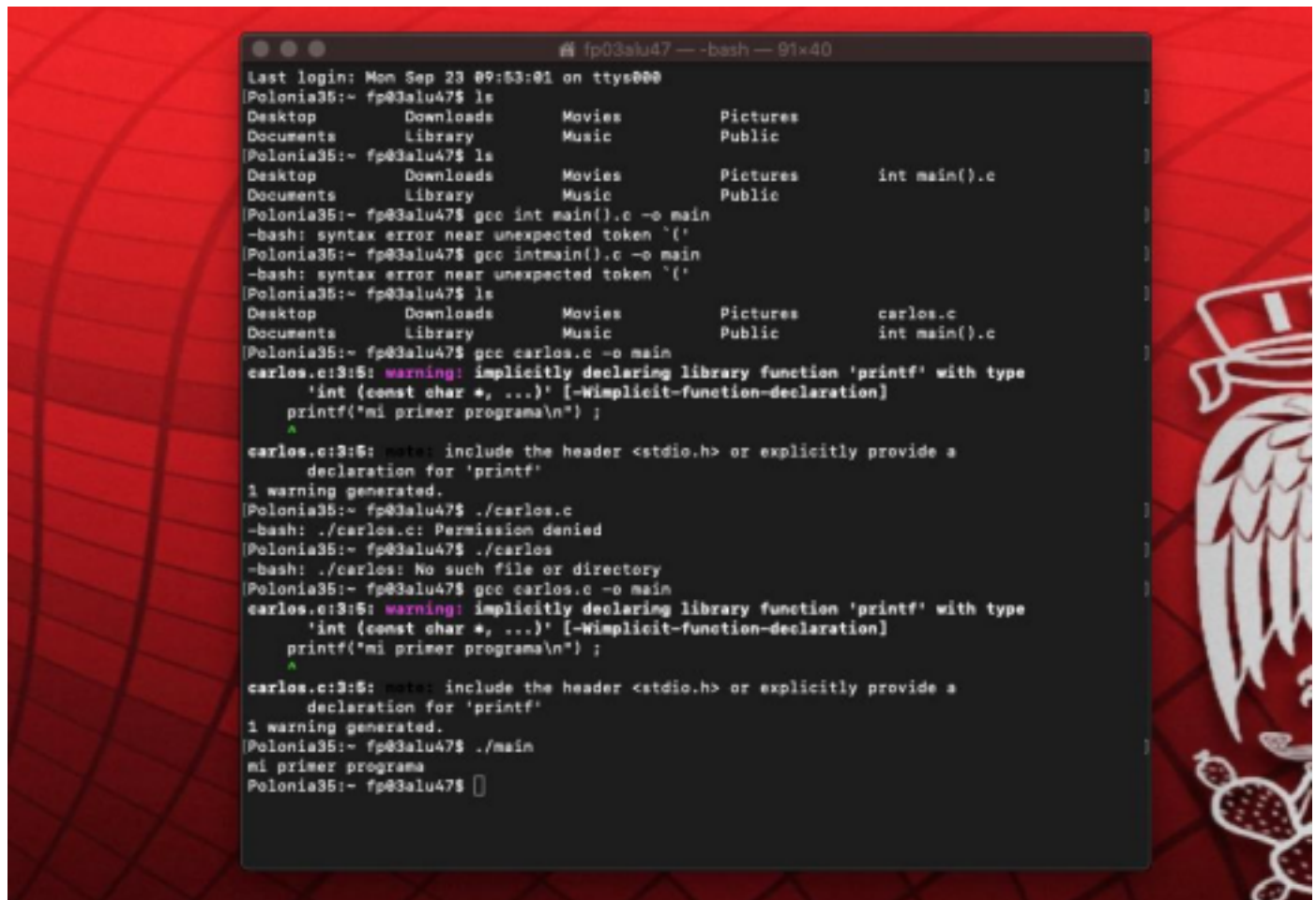
Pulsando una R mayúscula se entra en modo Replace hasta que, para salir, pulsamos <ESC>. Escribimos <<: set xxx>> fija la opción <<xxx>>.

NOTEPAD++:

Notepad++ es un editor gráfico mucho más parecido a lo que están acostumbrados.



Abrimos un programa en la maquina de trabajo que se llama sublime text con el cual empezamos a ver un tipo de código mas orientado a la programación en C, ya abierto este programa hicimos lo que nos decía la práctica tal cual, lo guardamos con cualquier nombre y al final “.c” para que saliera de colores.

A screenshot of a terminal window with a dark background and light text. The window title is "fp03alu47 - bash - 91x40". The terminal shows a series of commands and their outputs. It starts with a login message, then directory listings. The user attempts to compile a file named "int main().c" using "gcc", which fails with a syntax error. Then, the user tries to compile "intmain().c", which also fails. Next, the user lists files and finds "carlos.c" and "int main().c". They then compile "carlos.c" with "gcc carlos.c -o main", which produces a warning about an implicitly declared 'printf' function. The user then tries to run the compiled program with "./carlos.c", which fails with a permission denied error. They then try to run the executable with "./carlos", which fails with a "No such file or directory" error. Finally, they compile "carlos.c" again with "gcc carlos.c -o main", which also produces the same warning. They then run the executable with "./main", which successfully prints "mi primer programa" to the terminal.

```

Last login: Mon Sep 23 09:53:01 on ttys000
Polonia35:~ fp03alu47$ ls
Desktop      Downloads    Movies       Pictures
Documents    Library      Music        Public
Polonia35:~ fp03alu47$ ls
Desktop      Downloads    Movies       Pictures    int main().c
Documents    Library      Music        Public
Polonia35:~ fp03alu47$ gcc int main().c -o main
-bash: syntax error near unexpected token `{ '
Polonia35:~ fp03alu47$ gcc intmain().c -o main
-bash: syntax error near unexpected token `{ '
Polonia35:~ fp03alu47$ ls
Desktop      Downloads    Movies       Pictures    carlos.c
Documents    Library      Music        Public      int main().c
Polonia35:~ fp03alu47$ gcc carlos.c -o main
carlos.c:3:5: warning: implicitly declaring library function 'printf' with type
      'int (const char *, ...)' [-Wimplicit-function-declaration]
      printf("mi primer programa\n") ;
      ^
carlos.c:3:5: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
Polonia35:~ fp03alu47$ ./carlos.c
-bash: ./carlos.c: Permission denied
Polonia35:~ fp03alu47$ ./carlos
-bash: ./carlos: No such file or directory
Polonia35:~ fp03alu47$ gcc carlos.c -o main
carlos.c:3:5: warning: implicitly declaring library function 'printf' with type
      'int (const char *, ...)' [-Wimplicit-function-declaration]
      printf("mi primer programa\n") ;
      ^
carlos.c:3:5: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
Polonia35:~ fp03alu47$ ./main
mi primer programa
Polonia35:~ fp03alu47$
```

Ya después de eso compilamos el código que hicimos y si nos salía “warning” es que estaba bien.

## REPORTE DE PRÁCTICA:

En esta practica ya nos adentramos más a lo que es programación en C, ya tuvimos una vista previa sobre los editores, los programas que nos servirán para hacer nuestra prácticas en casa, también vimos nuevas formas de darle ordines a git, lo del vim tutor, nos sirve para hacer muchas cosas con el teclado, para añadir textos, ejecutar mandatos, para poner cosas que ya han sido borradas anteriormente, para sustituir caracteres, para borrar palabras en lineas o en frases. También vimos los significados de ordenadores de archivos como los son txt, markdown, html, LaTeX, csv. Al parecer esto se va poniendo cada vez más y más difícil, pero por lo pronto esto parece ser un poco fácil ya que estemos cada

vez más relacionados con este tipo de lenguaje.