	Carátula para entrega de prácticas	
Facultad de Ingeniería		Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): #6

Integrante(s): Torres Mendoza Alexa Erandy

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 49

Semestre: 2020-1

Fecha de entrega: 30/septiembre/2019

Observaciones: La práctica esta incompleta, le falta el desarrollo del editor vim.

CALIFICACIÓN: 7

- **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

- **Texto plano**

El texto plano es el tipo de archivo más sencillo que hay, ya que en su contenido no hay otra cosa mas que el texto que lo conforma.

- **Tipos de archivo**

- **TXT**- Un archivo de texto simple, es un archivo informático que contiene únicamente texto formado solo por caracteres que son legibles por humanos, careciendo de cualquier tipo de formato tipográfico. Representan caracteres ordinarios como letras, números y signos de puntuación.
- **MARKDOWN**- Markdown es un lenguaje de marcado ligero creado por John Gruber que trata de conseguir la máxima legibilidad y facilidad de publicación tanto en su forma de entrada como de salida, inspirándose en muchas convenciones existentes para marcar mensajes de correo electrónico usando texto plano.
- **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.
- **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 (del inglés comma-separated values) 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 (o punto y coma en donde la coma es el separador decimal).

- **Editores**

Un editor de texto es un programa que es capaz de editar texto plano. No confundir con los procesadores de texto.

- Atom
- Bluefish
- Brackets
- Gedit
- Geany

- Emacs
- Nano
- Notepad++
- Pico
- Sublime Text
- Vim

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

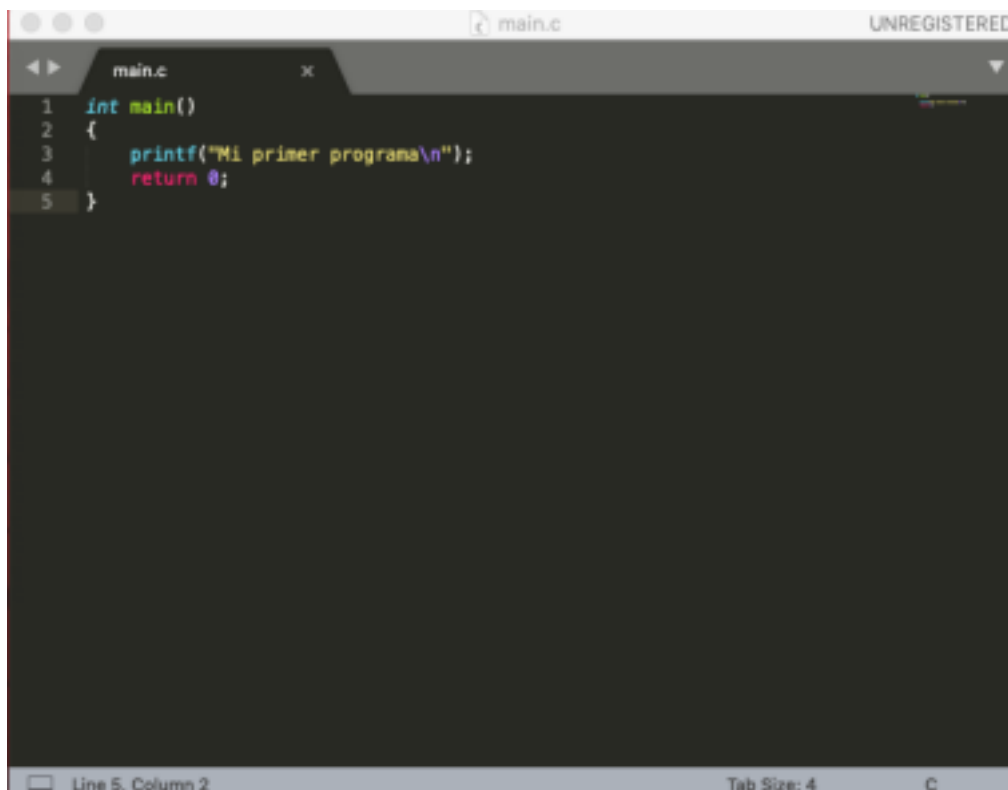
- Nano

Un editor de texto de terminal. Muy simple y básico, útil principalmente para editar textos cortos como los commits de git.

- 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.

ACTIVIDAD 2



The image shows a screenshot of a text editor window. The title bar at the top indicates the file is 'main.c' and the editor is 'UNREGISTERED'. The editor has a dark background with syntax-highlighted C code. The code is as follows:

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

At the bottom of the window, a status bar shows 'Line 5, Column 2' and 'Tab Size: 4'.

```
Documents — -bash — 80x24
[Noruega30:Documents fp03alu49$ gcc int main().c main_
-bash: syntax error near unexpected token `('
[Noruega30:Documents fp03alu49$ ls
int main().c
[Noruega30:Documents fp03alu49$ gcc int main().c -o main_
-bash: syntax error near unexpected token `('
[Noruega30:Documents fp03alu49$ ls
int main().c    main.c
[Noruega30:Documents fp03alu49$ gcc main.c -o main_
main.c:3:2: warning: implicit declaration of function 'print' is invalid in C99
      [-Wimplicit-function-declaration]
      print("Mi primer programa\n");
      ^
1 warning generated.
Undefined symbols for architecture x86_64:
  "_print", referenced from:
      _main in main-d025d0.o
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
[Noruega30:Documents fp03alu49$ gcc main.c -o main
main.c:3:2: warning: implicit declaration of function 'print' is invalid in C99
      [-Wimplicit-function-declaration]
      print("Mi primer programa\n");
      ^
```

```
Documents — -bash — 80x24
Last login: Mon Sep 23 09:05:16 on console
[Noruega30:~ fp03alu49$ nano
[Noruega30:~ fp03alu49$ nano
[Noruega30:~ fp03alu49$ cd
.CFUserTextEncoding Desktop/      Movies/
.DS_Store             Documents/    Music/
.Trash/               Downloads/   Pictures/
.bash_sessions/       Library/     Public/
[Noruega30:~ fp03alu49$ cd Downloads/
[Noruega30:Downloads fp03alu49$ cd .localized
-bash: cd: .localized: Not a directory
[Noruega30:Downloads fp03alu49$ cd
[Noruega30:~ fp03alu49$ cd
.CFUserTextEncoding Desktop/      Movies/
.DS_Store             Documents/    Music/
.Trash/               Downloads/   Pictures/
.bash_sessions/       Library/     Public/
[Noruega30:~ fp03alu49$ cd Documents/
[Noruega30:Documents fp03alu49$ cd
.localized    int main().c
[Noruega30:Documents fp03alu49$ ls
int main().c
[Noruega30:Documents fp03alu49$ gcc main.c -o main_
clang: error: no such file or directory: 'main.c'
```

```

Documents — -bash — 80x24
[Noruega30:Documents fp03alu49$ cd
.localized int main().c
[Noruega30:Documents fp03alu49$ ls
int main().c
[Noruega30:Documents fp03alu49$ gcc main.c -o main_
clang: error: no such file or directory: 'main.c'
clang: error: no input files
[Noruega30:Documents fp03alu49$ ls
int main().c
[Noruega30:Documents fp03alu49$ gcc int main.c -o main_
clang: error: no such file or directory: 'int'
clang: error: no such file or directory: 'main.c'
clang: error: no input files
[Noruega30:Documents fp03alu49$ ls
int main().c
[Noruega30:Documents fp03alu49$ gcc int main().c main_
-bash: syntax error near unexpected token `{ '
[Noruega30:Documents fp03alu49$ ls
int main().c
[Noruega30:Documents fp03alu49$ gcc int main().c -o main_
-bash: syntax error near unexpected token `{ '
[Noruega30:Documents fp03alu49$ ls
int main().c main.c
[Noruega30:Documents fp03alu49$ gcc main.c -o main_

```

```

Documents — -bash — 80x24
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
[Noruega30:Documents fp03alu49$ gcc main.c -o main
main.c:3:5: warning: implicit declaration of function 'print' is invalid in C99
      [-Wimplicit-function-declaration]
    print("Mi primer programa\n");
    ^
1 warning generated.
Undefined symbols for architecture x86_64:
  "_print", referenced from:
    _main in main-252566.o
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
[Noruega30:Documents fp03alu49$ gcc main.c -o main
main.c:3:5: warning: implicitly declaring library function 'printf' with type
      'int (const char *, ...)' [-Wimplicit-function-declaration]
    printf("Mi primer programa\n");
    ^
main.c:3:5: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
1 warning generated.
[Noruega30:Documents fp03alu49$ ./main
Mi primer programa
[Noruega30:Documents fp03alu49$

```

CONCLUSIONES

Es necesario conocer que es un texto plano y para qué sirve, así como, los tipos de archivo y editores, ya que al momento de ejecutarlos tienen una funcionalidad y propósito en diferentes casos saber usarlos me facilita a mí como alumna poder realizar el protocolo correspondiente para programar y verificar un proceso determinado en lenguaje C en diversos sistemas operativos.