

## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Profesor:	César Fabián Domínguez Velasco
Asignatura:	Fundamentos de Programación
No. de práctica(s):	06
Integrante(s):	12_García_Sánchez_Alejandro 17_Lopez_Castro_Anastacia 32_Ramirez_Rivas_Gael 37_Ruiz_Hernandez_Ruben_Antonio
No. de lista o brigada:	1A
Semestre:	2024-2
Fecha de entrega:	20 de marzo del 2024
Observaciones:	
	CALIFICACIÓN:

# PRÁCTICA 06: ENTORNO Y FUNDAMENTOS DEL LENGUAJE C

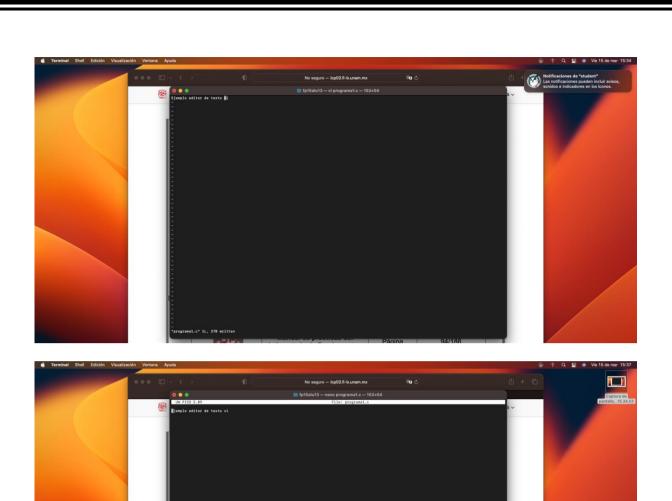
#### 1) Objetivo:

El alumno elaborará programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

#### 2) Introducción:

Una vez que un problema dado ha sido analizado (se identifican los datos de entrada y la salida deseada), que se ha diseñado un algoritmo que lo resuelva de manera eficiente (procesamiento de datos), y que se ha representado el algoritmo de manera gráfica o escrita (diagrama de flujo o pseudocódigo) se puede proceder a la etapa de codificación, ya que se puede realizar en cualquier lenguaje de programación estructurada, de igual manera a lo largo de este curso se aprenderá el uso del lenguaje de programación C, el cual es de nivel medio.

- 3) Desarrollo (Capturas de pantalla de los pseudocódigos):
- Introducción de manejo mediante la terminal Cygwin en una computadora Mac:



### ■ Programa 1 en Visual Studio Code:

"R Read File

^Y Prev Pg ^Y Next Pg Cut Text
UnCut Text

Cur Pos

#### Ejecución del Programa 1:

```
Late login: Fri Mar 15 15:19:54 om ttys000

The default interactive shall is now zsh.
To update your account to use zsh, plasee run 'chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT288050.

Alemania03:- fp15slu13$ vi programs1.c

Alemania03:- fp15slu13$ nnop programs1.c

Alemania03:- fp15slu13$ nnop programs1.c

Alemania03:- fp15slu13$ and documents/
Alemania03:- fp15slu13$ ls

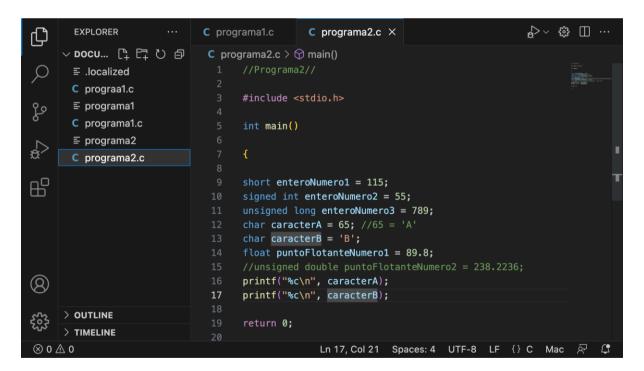
programs1.c

Alemania03:- fp15slu13$ ls

programs1.c

Alemania03:documents fp15slu13$ |
```

#### Programa 2 en Visual Studio Code:



```
Andorra04:documents fp15alu39$ gcc programa2.c -o programa2
Andorra04:documents fp15alu39$ ./programa2
A
B
Andorra04:documents fp15alu39$ ■
```

#### Programa 3 en Visual Studio Code:

```
C programa2.c
                                                                           C programa3.c ×
                                                                                                                                                                                                                                           $>∨ ∰ Ш ···
           C programa1.c
                        #include <stdio.h>
                        int main ()
                         int entero;
品
                         double doble:
                         entero = 14;
                         doble = 6.8e10:
                        //Funciones de salida de datos en pantalla
printf("La variable entera tiene valor: %i \n", entero);
printf("La variable flotante tiene valor: %f \n", flotante);
               19
                        printf("La variable doble tiene valor: %f \n", doble);
printf("La variable caracter tiene valor: %c \n", caracter);
printf("Entero como octal: %o \n Como Hexadecimal %X \n", entero, entero);
printf("Flotante con precisión: %5.2f \n", flotante);
printf("Doble con precisión: %5.2f \n", doble);
                         printf("Carácter como entero: %d \n", caracter);
(2)
563
                                                                                                                                                                         Ln 19, Col 48 Spaces: 4 UTF-8 LF {} C Mac 🔊 🚨
```

Andorra04:documents fp15alu39\$ gcc programa3.c -o programa3

Andorra04:documents fp15alu39\$ ./programa3

La variable entera tiene valor: 14

La variable flotante tiene valor: 3.500000

La variable doble tiene valor: 68000000000.000000

La variable caracter tiene valor: A

Entero como octal: 16

Como Hexadecimal E

Flotante con precisión: 3.50

Doble con precisión: 68000000000.00

Carácter como entero: 65

Andorra04:documents fp15alu39\$

#### Programa 4 en Visual Studio Code:

```
EXPLORER
                                                            C programa3.c
                                                                                  C programa4.c × ♣> ✓ ∰ Ⅲ ···
ф
                                     C programa4.c > 😭 main()
        c prograa1.c
                                            #include <stdio.h>
        ≡ programa1
حړ
        c programa1.c
        ≡ programa2
        c programa2.c
        ≡ programa3
                                             int entero:
品
        c programa3.c
                                             float flotante;
        c programa4.c
                                             print("%i", &entero);
printf("El valor ingresado es: %d\n", entero);
printf("Ingresa el valor float: ");
                                             scanf("%f", &flotante);
printf("El valor ingresado es: %f\n", flotante);
      > OUTLINE
                                             return 0;
      > TIMELINE
⊗ 0 △ 0
                                                             Ln 18, Col 1 Spaces: 4 UTF-8 LF {} C
```

```
Andorra04:documents fp15alu39$ clear

Andorra04:documents fp15alu39$ gcc programa4.c -o programa4

Andorra04:documents fp15alu39$ ./programa4

Ingresa el valor entero: 14

El valor ingresado es: 14

Ingresa el valor float: 15.5

El valor ingresado es: 15.500000
```

#### Programa 5 en Visual Studio Code:

Andorra04:documents fp15alu39\$

```
DOUGERS

C C Inclamundo.

C Swap.

C C Inclamundo.

C Swap.

C C Calculadora.

C ProgramaSc

C C Calculadora.

C C ProgramaSc

C C Calculadora.

C C Codecision.

C C ComparmaSc

C C None

C None
```

#### Programa 6 en Dev C++:

```
//Compila y ejecuta bien
       //Programa6.c
       #include <stdio.h>
       int main()
            short ocho, cinco, cuatro, tres, dos, uno;
            //8 en binario: 0000 0000 0000 1000
            ocho = 8;
            //5 en binario: 0000 0000 0000 0101
            cinco = 5;
            //4 en binario: 0000 0000 0000 0100
            cuatro = 4;
            //3 en binario: 0000 0000 0000 0011
            tres = 3;
            //2 en binario: 0000 0000 0000 0010
            dos = 2;
            //1 en binario: 0000 0000 0000 0001
            uno = 1;
            printf(" Operadores aritméticos\n");
            pringtf(" 5 modulo 2 = %d\n", cinco%dos);
            printf( " Operadores lógicos\");
            printf(" 8>>2=%d\n", ocho>>dos);
            printf(" 5 & 4 =%d\n", cinco&cuatro);
            printf(" 3| 2=%d\n", tres|dos);
            printf("\n");
            return 0;
       }
//Compila y ejecuta bien
//Programa6.c
include <stdio.h>
int main()
   short ocho, cinco, cuatro, tres, dos, uno;
   //8 en binario: 0000 0000 0000 1000
   ocho = 8;
                                                  Compile Progress
   //5 en binario: 0000 0000 0000 0101
                                                  Progress Log
   cinco = 5;
   //4 en binario: 0000 0000 0000 0100
                                                   Compiler: Default compiler
   cuatro = 4:
   //3 en binario: 0000 0000 0000 0011 tres = 3;
                                                   Status:
                                                        Done.
                                                   File:
   //2 en binario: 0000 0000 0000 0010
   dos = 2;
                                                   Errors:
                                                                0 Warnings:
                                                                                0
   //1 en binario: 0000 0000 0000 0001
   uno = 1:
   printf(" Operadores aritméticos\n");
                                                                 <u>C</u>lose
   pringtf(" 5 modulo 2 = %d\n", cinco%dos);
printf( " Operadores lógicos\");
   printf(" 8>>2=%d\n", ocho>>dos);
   printf(" 5 & 4 =%d\n",cinco&cuatro);
   printf(" 3| 2=%d\n", tres|dos);
   printf("\n");
   return 0;
```

```
A X 登 湯
```

```
Operadores aritméticos
5 modulo 2 = 1
Operadores lógicos
8>>2=2
5 & 4 = 4
3 | 2=3

...Program finished with exit code 0
Press ENTER to exit console.
```

#### 4) Conclusiones (Individuales):

#### ■ García Sánchez Alejandro:

Realizar esta práctica me ayudó a comprender qué es y la función de un archivo de texto, además el repaso de comando y explorar un poco más Visual Studio Code acompañado del uso de la terminal.

En general adquirí los conocimientos requeridos por la práctica y los ejercicios propuestos en la misma.

- López Castro Anastacia: Al realizar esta práctica, se nos facilitó un poco más debido a clases anteriores ya que en estas, conocimos poco a poco el lenguaje de programación.Lo que se me complicó más de esta práctica fue la compilación en gcc, ya que me costaba poder guardar el documento y después poder compilarlo en Cygwin, sin embargo, con el apoyo del profesor logré compilar sin ninguna dificultad.
- Ramírez Rivas Gael: En esta práctica se realizaron actividades orientadas a conocer el lenguaje de programación C, tales como crear programas, definir variables y crear algoritmos. Estas actividades se desarrollaron con ayuda del editor de código Visual Studio Code. El desarrollo de estas actividades presentó algunas dificultades en función

de nuestra familiarización y experiencia previa en esta forma de programar, sin embargo, con práctica y ayuda para la solución de dudas, se pudieron completar las actividades sin mayores inconvenientes cumpliendo los objetivos planteados.

Ruíz Hernández Rubén Antonio: Comprendí al pie de la letra la diferencia entre compilar y ejecutar un programa, ya que recopilar se refiere al proceso de convertir el código fuente de un programa en lenguaje de programación durante un archivo ejecutable, mientras que la ejecución es el proceso de inicio y correr el programa compilado, permitiendo que se realicen las instrucciones u operaciones especificadas en el código fuente, de igual manera este objetivo se puso en práctica mediante la realización de programas en lenguaje C mediante la aplicación del Visual Studio Code y Dev C++, junto con la terminal Cygwin mediante computadoras Mac.

#### 5) Retroalimentación (Equipo):

Siguiendo las instrucciones de la práctica, se utilizó una terminal y la herramienta de edición de código Visual Studio Code y Dev C++ para la realización de actividades, las cuales consistieron en elaborar programas en lenguaje C, utilizando el lenguaje incluyendo la corrección de sintaxis adecuada para evitar errores, además de tener acercamiento a procesos como la definición de variables. En esta práctica se buscó la familiarización con el lenguaje C a partir de un acercamiento orientado a conocer aspectos generales de este lenguaje de programación. Además de solucionar errores que se presentaron en la ejecución, así como resolver dudas, permitiéndonos completar los objetivos de esta para generar un conocimiento útil para aplicarlo según las necesidades de nuestras carreras.

#### 6) Fuentes en APA:

Laboratorio Salas A y B. (s.f.). Manual de Prácticas de la Asignatura Fundamentos de Programación (Guía práctica de estudio 06:

Entorno y Fundamentos del Lenguaje C, pág. 88 - 114). Recuperado el 15 de Marzo del 2024, de Laboratorio de Computación Salas A y B: http://lcp02.fi-b.unam.mx/ RuyAntonio. (s.f.). GitHub - RuyAntonio/practica6\_fdp: Práctica 06: Entorno y Fundamentos del Lenguaje C, de GitHub. Recuperado el 15 de Marzo del 2024, de GitHub: https://github.com/RuyAntonio/practica6\_fdp.git