



Taller Ejercicios C

Computación Paralela Distribuida

Daniel Alejandro Olarte Ávila

Universidad Sergio Arboleda

Universidad Sergio Arboleda Cl. 74 #14-14

Bogotá, Colombia

Correo: danielolarte2012@gmail.com

Escuela de Ciencias Exactas e Ingeniería

Profesor: John Corredor

19/08/2022

TALLER EJERCICIOS COMPUTACIÓN PARALELA DISTRIBUIDA

RESUMEN:

En este taller se realizarán 10 ejercicios de C, cada uno con un problema diferente, donde se podrá probar diferentes habilidades donde cada vez va aumentando la dificultad a la hora de programar y así poder demostrar y aprender el conocimiento adquirido en este lenguaje, se realizará los ejercicios por terminal en Linux, se mostrara como se crean con comandos los archivos, como se modifican y su respectiva salida.

INTRODUCCIÓN:

Se realizará el taller en el lenguaje de programación c, el cual es un lenguaje fuertemente tipado, y se lleva muy bien con la programación orientada a objetos, en el taller se realizará desde ejercicios básicos hasta unos más complejos, abarcando así diferentes opciones que se puede realizar en el lenguaje para cumplir el objetivo de cada programa

PRIMER PASO: Abrir la terminal para crear cada ejercicio del taller

```
daniel@daniel-VirtualBox: $
```

SEGUNDO PASO: Nos direccionamos a la ruta en donde vamos a crear los proyectos

```
daniel@daniel-VirtualBox:~$ cd Escritorio/  
daniel@daniel-VirtualBox:~/Escritorio$ cd Trabajos\ U/  
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U$ cd Paralela/  
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela$ cd PrimerTallerC/  
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

CREACION DE LOS PROYECTOS:

Primer Ejercicio:

CREACIÓN DEL PROYECTO: Se escribe el comando “nano primerEjercicio.c” para crear el archivo de c y empezar a escribir el código

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano primerEjercicio.c
```



The screenshot shows the GNU nano 6.2 text editor interface. The title bar at the top indicates the file is 'ejercicio1.cpp'. The editor area is currently blank. At the bottom, there is a status bar with various keyboard shortcuts for editing, such as 'Ayuda', 'Guardar', 'Buscar', 'Cortar', 'Ejecutar', 'Ubicación', 'Desahacer', 'Poner marca', 'A llave', 'Anterior', 'Atrás', and 'Palabr ant'.

CÓDIGO:

```
GNU nano 6.2 primerEjercicio.c
/**
 * @file primerEjercicio.c
 * @author Daniel Olarte
 * @brief producto de 2 numeros
 * @version 0.1
 * @date 2022-08-16
 *
 * @copyright Copyright (c) 2022
 */

/*
1.- Escribe un programa en C que acepte dos enteros del usuario y calcule el producto de los
dos enteros.
*/

#include <stdio.h>
int main()
{
    int num1, num2, producto;
    printf("Introduzca el primer entero: ");
    scanf("%d", &num1);
    printf("Introduzca el segundo entero: ");
    scanf("%d", &num2);
    producto = num1 * num2;
    printf("Producto de los dos enteros anteriores = %d", producto);
    return 0;
}
```

CREACIÓN DEL EJECUTABLE: Se crea el ejecutable con el comando “gcc (nombre del archivo) -o (nombre del ejecutable)”

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc primerEjercicio.c -o primero
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Se ejecuta el comando “./(nombre del ejecutable)” para compilar el proyecto

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc primerEjercicio.c -o primero
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ls
cuartoEjercicio.c  novenoEjercicio.c  primerEjercicio.c  quintoEjercicio.c  septimoEjercicio.c  tercerEjercicio.c
decimoEjercicio.c  octavoEjercicio.c  primero            segundoEjercicio.c  sextoEjercicio.c
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./primero
Introduzca el primer entero: 5
Introduzca el segundo entero: 15
Producto de los dos enteros anteriores = 75
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Segundo Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano segundoEjercicio.c
```

CÓDIGO:

```
GNU nano 6.2                                segundoEjercicio.c
/**
 * @file segundoEjercicio.c
 * @author Daniel Olarte
 * @brief convertir años, semanas y días
 * @version 0.1
 * @date 2022-08-16
 *
 *
 * @copyright Copyright (c) 2022
 *
 */

/*
    Escribe un programa en C para convertir los días especificados en años, semanas y días.
*/

#include <stdio.h>
int main()
{
    int dias, semanas, anios;
    printf("Introduzca los días: ");
    scanf("%d", &dias);
    anios = dias / 365;
    semanas = (dias % 365) / 7;
    dias = (dias % 365) % 7;
    printf("%d días equivalen a %d años, %d semanas y %d días", dias, anios, semanas, dias);
    return 0;
}
```

CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc segundoEjercicio.c -o segundo
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./segundo
Introduzca los días: 1928
5 días equivalen a 5 años, 14 semanas y 5 días
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Tercer Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano tercerEjercicio.c
```

CÓDIGO:

```
GNU nano 6.2                                     tercerEjer
/**
 * @file tercerEjercicio.c
 * @author Daniel Olarte
 * @brief distancia entre 2 puntos
 * @version 0.1
 * @date 2022-08-16
 *
 * @copyright Copyright (c) 2022
 */

/*
Escribe un programa en C para calcular la distancia entre los dos puntos.
*/
#include <stdio.h>

//funcion para hallar la raiz cuadrada manualmente
float raiz(float numero){
    float resultado = 0;
    int i = 0;
    while(resultado*resultado < numero){
        resultado += 0.1;
        i++;
    }
    return resultado;
}

int main()
{
    int x1, x2, y1, y2;
    printf("Introduce la coordenada x del primer punto: ");
    scanf("%d", &x1);
    printf("Introduce la coordenada y del primer punto: ");
    scanf("%d", &y1);
    printf("Introduce la coordenada x del segundo punto: ");
    scanf("%d", &x2);
    printf("Introduce la coordenada y del segundo punto: ");
    scanf("%d", &y2);
    float result = raiz((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    printf("La distancia entre los dos puntos es: %f", result);

    return 0;
}
```

CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc tercerEjercicio.c -o tercero
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./tercero
Introduce la coordenada x del primer punto: 15
Introduce la coordenada y del primer punto: 20
Introduce la coordenada x del segundo punto: 5
Introduce la coordenada y del segundo punto: 12
La distancia entre los dos puntos es: 12.900013
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Cuarto Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano cuartoEjercicio.c
```

CÓDIGO:

```
GNU nano 6.2                                                                    cuartoEje
/**
 * @file cuartoEjercicio.c
 * @author Daniel Olarte
 * @brief 10 primeros numeros
 * @version 0.1
 * @date 2022-08-16
 *
 * @copyright Copyright (c) 2022
 *
 */

/*
    Escribe un programa en C para mostrar los 10 primeros números naturales.
*/
#include <stdio.h>

int main()
{
    int i = 0;
    while(i < 10){
        printf("%d\n", i+1);
        i++;
    }
    return 0;
}
```


CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc cuartoEjercicio.c -o cuarto
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./cuarto
1
2
3
4
5
6
7
8
9
10
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Quinto Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano quintoEjercicio.c
```

CÓDIGO:

```
/**
 * @file quintoEjercicio.c
 * @author Daniel Olarte
 * @brief triangulo rectangulo con asteriscos
 * @version 0.1
 * @date 2022-08-16
 *
 *
 * @copyright Copyright (c) 2022
 */

/*
Escribe un programa en C para mostrar el patrón como triángulo rectángulo usando un
asterisco. La cantidad de líneas del triángulo es ingresada por consola por el usuario, y el valor
máximo de líneas es de 14.
*/

#include <stdio.h>

int main()
{
    //declaracion de variables
    int lineas;
    //pedir al usuario la cantidad de lineas
    printf("Introduce la cantidad de lineas del triangulo: ");
    scanf("%d", &lineas);
    //maximo 14 lineas
    if (lineas > 14){//si el numero de lineas es mayor a 14
        printf("El numero de lineas no puede ser mayor a 14");
    }
    else{//si el numero de lineas es menor a 14

        int i = 0;
        //bucle para mostrar el triangulo
        while(i < lineas){
            int j = 0;
            while(j < i+1){
                printf("*");
                j++;
            }
            printf("\n");
            i++;
        }

        return 0;
    }
}
```

CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc quintoEjercicio.c -o quinto
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./quinto
Introduce la cantidad de líneas del triángulo: 8
*
**
***
****
*****
*****
*****
*****
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Sexto Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano sextoEjercicio.c
```

CÓDIGO:

```
GNU nano 6.2                                                                    sextoEjercicio.c
/**
 * @file sextoEjercicio.c
 * @author Daniel Olarte
 * @brief matriz de 1's
 * @version 0.1
 * @date 2022-08-16
 * @copyright Copyright (c) 2022
 */
/*
Escribe un programa que imprima una matriz de "1", de tamaño NxN. El tamaño es
indicado por el usuario y debe ser siempre menor que 8.
*/

#include <stdio.h>

int main()
{
    int n, i, j;
    printf("Introduce el tamaño de la matriz: ");
    scanf("%d", &n);
    if (n > 8){
        printf("El tamaño de la matriz no puede ser mayor a 8");
    }
    else{
        for (i = 0; i < n; i++){
            for (j = 0; j < n; j++){
                printf("1 ");
            }
            printf("\n");
        }
    }
    return 0;
}
```

CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc sextoEjercicio.c -o sexto
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./sexto
Introduce el tamaño de la matriz: 5
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Séptimo Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano septimoEjercicio.c
```

CÓDIGO:

```
GNU nano 6.2                                     septimoEjercicio.c *
/**
 * @file septimoEjercicio.c
 * @author Daniel Olarte
 * @brief matriz de aleatorios
 * @version 0.1
 * @date 2022-08-16
 * @copyright Copyright (c) 2022
 *   Escribe un programa que imprima una matriz de números aleatorios enteros, entre 0 y 9, de
 *   de tamaño NxN. El tamaño es indicado por el usuario y debe ser siempre menor que 8.
 */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int n, i, j;
    printf("Introduce el tamaño de la matriz: ");
    scanf("%d", &n);
    if (n > 8){
        printf("El tamaño de la matriz no puede ser mayor a 8");
    }
    else{
        srand(time(NULL));
        for (i = 0; i < n; i++){
            for (j = 0; j < n; j++){
                printf("%d ", rand() % 10);
            }
            printf("\n");
        }
    }
    return 0;
}
```

CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc septimoEjercicio.c -o septimo
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./septimo
Introduce el tamaño de la matriz: 5
8 2 6 6 0
3 6 5 6 6
5 7 6 7 1
2 4 3 0 1
2 3 9 4 6
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Octavo Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano octavoEjercicio.c
```

CÓDIGO:

```
GNU nano 6.2 octavoEjercicio.c *
/**
 * @file octavoEjercicio.c
 * @author Daniel Olarte
 * @brief suma de matriz de aleatorios
 * @version 0.1
 * @date 2022-08-16
 * @copyright Copyright (c) 2022
 * Escribe un programa que muestre dos matrices de números aleatorios, entre 0 y 9, de
 * tamaño NxN. El tamaño es indicado por el usuario y debe ser siempre menor que 8.
 * Finalmente, se presenta la suma de las dos matrices.
 */

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, i, j;
    printf("Introduce el tamaño de la matriz: ");
    scanf("%d", &n);
    if (n > 8){
        printf("El tamaño de la matriz no puede ser mayor a 8");
    }
    else{
        //crea la matriz
        int matriz1[n][n];
        int matriz2[n][n];
        //rellena la matriz
        for (i = 0; i < n; i++){
            for (j = 0; j < n; j++){
                matriz1[i][j] = rand() % 10;
                matriz2[i][j] = rand() % 10;
            }
        }
    }
}
```

```
GNU nano 6.2 octavoEjercicio.c *
}

//suma las matrices
int suma[n][n];
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        suma[i][j] = matriz1[i][j] + matriz2[i][j];
    }
}

//muestra las matrices
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", matriz1[i][j]);
    }
    printf("\n");
}
printf("\n");
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", matriz2[i][j]);
    }
    printf("\n");
}
printf("\n");
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", suma[i][j]);
    }
    printf("\n");
}
return 0;
}
```

CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc octavoEjercicio.c -o octavo
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./octavo
Introduce el tamaño de la matriz: 4
3 7 3 6
9 2 0 3
0 2 1 7
2 2 7 9

6 5 5 2
1 7 9 6
6 6 8 9
0 3 5 2

9 12 8 8
10 9 9 9
6 8 9 16
2 5 12 11
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

Noveno Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano novenoEjercicio.c
```

CÓDIGO:

```
GNU nano 6.2 novenoEjercicio.c *
/**
 * @file novenoEjercicio.c
 * @author Daniel Olarte
 * @brief resta de matriz de aleatorios
 * @version 0.1
 * @date 2022-08-16
 * @copyright Copyright (c) 2022
 * Escribe un programa que muestre dos matrices de números aleatorios, entre 0 y 9, de
 * tamaño NxN. El tamaño es indicado por el usuario y debe ser siempre menor que 8.
 * Finalmente, se presenta la resta de las dos matrices.
 */

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, i, j;
    printf("Introduce el tamaño de la matriz: ");
    scanf("%d", &n);
    if (n > 8){
        printf("El tamaño de la matriz no puede ser mayor a 8");
    }
    else{
        //crea la matriz
        int matriz1[n][n];
        int matriz2[n][n];
        //rellena la matriz
        for (i = 0; i < n; i++){
            for (j = 0; j < n; j++){
                matriz1[i][j] = rand() % 10;
                matriz2[i][j] = rand() % 10;
            }
        }
    }
}
```

```
//resta las matrices
int resta[n][n];
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        resta[i][j] = matriz1[i][j] - matriz2[i][j];
    }
}

//muestra las matrices
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", matriz1[i][j]);
    }
    printf("\n");
}
printf("\n");
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", matriz2[i][j]);
    }
    printf("\n");
}
printf("\n");
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", resta[i][j]);
    }
    printf("\n");
}
```

```
} }
```

CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc novenoEjercicio.c -o noveno
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./novenoejercicio
Introduce el tamaño de la matriz: 4
3 7 3 6
9 2 0 3
0 2 1 7
2 2 7 9

6 5 5 2
1 7 9 6
6 6 8 9
0 3 5 2

-3 2 -2 4
8 -5 -9 -3
-6 -4 -7 -2
2 -1 2 7
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```


Decimo Ejercicio:

CREACIÓN DEL PROYECTO:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ nano decimoEjercicio.c
```

CÓDIGO:

```
GNU nano 6.2 decimoEjercicio.c *
/**
 * @file decimoEjercicio.c
 * @author Daniel Olarte
 * @brief producto de matriz de aleatorios
 * @version 0.1
 * @date 2022-08-16
 * @copyright Copyright (c) 2022
 * Escribe un programa que muestre dos matrices de números aleatorios, entre 0 y 9, de
 * tamaño NxN. El tamaño es indicado por el usuario y debe ser siempre menor que 8.
 * Finalmente, se presenta el producto de las dos matrices.
 */

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, i, j;
    printf("Introduce el tamaño de la matriz: ");
    scanf("%d", &n);
    if (n > 8){
        printf("El tamaño de la matriz no puede ser mayor a 8");
    }
    else{
        //crea la matriz
        int matriz1[n][n];
        int matriz2[n][n];
        //rellena la matriz
        for (i = 0; i < n; i++){
            for (j = 0; j < n; j++){
                matriz1[i][j] = rand() % 10;
                matriz2[i][j] = rand() % 10;
            }
        }
    }
}
```

```
//producto de las matrices
int producto[n][n];
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        producto[i][j] = matriz1[i][j] * matriz2[i][j];
    }
}

//muestra las matrices
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", matriz1[i][j]);
    }
    printf("\n");
}
printf("\n");
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", matriz2[i][j]);
    }
    printf("\n");
}
printf("\n");
for (i = 0; i < n; i++){
    for (j = 0; j < n; j++){
        printf("%d ", producto[i][j]);
    }
    printf("\n");
}
return 0;
}
```

CREACIÓN DEL EJECUTABLE Y SALIDA:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ gcc decimoEjercicio.c -o decimo
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ./decimo
Introduce el tamaño de la matriz: 4
3 7 3 6
9 2 0 3
0 2 1 7
2 2 7 9

6 5 5 2
1 7 9 6
6 6 8 9
0 3 5 2

18 35 15 12
9 14 0 18
0 12 8 63
0 6 35 18
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

VISTA DE ARCHIVOS CREADOS:

```
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$ ls
cuarto      decimoEjercicio.c  octavo      primero      segundo      septimoEjercicio.c  tercerEjercicio.c
cuartoEjercicio.c  noveno      octavoEjercicio.c  quinto      segundoEjercicio.c  sexto      tercero
decimo      novenoEjercicio.c  primerEjercicio.c  quintoEjercicio.c  septimo      sextoEjercicio.c
daniel@daniel-VirtualBox:~/Escritorio/Trabajos U/Paralela/PrimerTallerC$
```

CONCLUSION:

Al realizar los ejercicios pudimos demostrar las habilidades con el lenguaje, desde operadores aritméticos, ciclos, métodos, hasta menús para tomar diferentes decisiones diferentes y que se pueda repetir el código, a través de esto pudimos evidenciar ejercicios que nos hicieron pensar y ser creativos para llegar a que el programa cumpla su función, ya sea operaciones normales, como también operaciones con matrices como sumas, restas y multiplicaciones entre ellas y mostrar su respectiva salida, además de esto perfeccionamos la habilidad con los comandos de Linux para realizar cada uno de los ejercicios, y crear sus ejecutables para así poder compilar exitosamente cada uno de los ejercicios.