



# Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA



## *Reporte de actividad*

**Materia:**

Lenguaje C

**Tema:**

Práctica No. 6: Modular Programa.

**Alumno:**

Joshua Osorio Osor...

**Matrícula:**

1293271

**Docente:**

Fernando Elihonai Saucedo Lares

**Fecha:**

13/03/2025

# ÍNDICE

<a href="#">Objetivo.....</a>	<a href="#">2</a>
<a href="#">Introducción.....</a>	<a href="#">2</a>
<a href="#">Lista de materiales.....</a>	<a href="#">2</a>
<a href="#">Desarrollo.....</a>	<a href="#">2</a>
<a href="#">Conclusiones.....</a>	<a href="#">2</a>
<a href="#">Bibliografía.....</a>	<a href="#">3</a>

## Objetivo

El objetivo de este ejercicio es que los estudiantes adquieran habilidades prácticas en el manejo de estructuras de datos como arreglos unidimensionales y matrices bidimensionales en un lenguaje de programación.

## Introducción

Crear un arreglo unidimensional de 5 posiciones que almacene los nombres de 5 alumnos.  
Implementar una matriz de 5x5 en la que cada fila corresponderá a un alumno, y cada columna almacenará su calificación en una de las 5 unidades.

Crear un segundo arreglo unidimensional de 5 posiciones que almacenará el promedio de las 5 unidades para cada alumno. El programa debe calcular automáticamente los promedios a partir de las calificaciones almacenadas en la matriz.

Mostrar en pantalla una tabla que incluya:

- Los nombres de los alumnos.
- Las calificaciones de las 5 unidades.
- El promedio final de cada alumno.

### Lista de materiales

- Computadora
- software: mingw

# Desarrollo

Codigo:

C/C++

```
/*-----
-LABORATORIO: 7 Matrices y arreglos.
-Nombre: Joshua Osorio
-Materia: LENGUAJE C 531
-Fecha: Marzo/12/2025
```

Descripción del Ejercicio:

Crear un arreglo unidimensional de 5 posiciones que almacene los nombres de 5 alumnos.

Implementar una matriz de 5x5 en la que cada fila corresponderá a un alumno, y cada columna almacenará su calificación en una de las 5 unidades.

Crear un segundo arreglo unidimensional de 5 posiciones que almacenará el promedio de las 5 unidades para cada alumno. El programa debe calcular automáticamente los promedios a partir de las calificaciones almacenadas en la matriz.

Mostrar en pantalla una tabla que incluya:

- Los nombres de los alumnos.
- Las calificaciones de las 5 unidades.
- El promedio final de cada alumno.

```
-----*/
```

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

```
#define numAlum 5
#define unidades 5
```

```
void AsignarNombres(char[][50]);
void AsignarCalificaciones(char nombreAlumno[numAlum][50], float
calificaciones[numAlum][unidades], float promedio[numAlum]);
void CalcularPromedio(char nombreAlumno[][50], float
calificaciones[numAlum][unidades], float promedio[numAlum]);
void MostrarInformacion(char nombreAlumno[][50], float
calificaciones[numAlum][unidades], float promedio[numAlum]);
```

```

int main(void)
{
    /* Semilla para la generacion de numeros aleatorios. */
    srand(time(NULL));

    int opc = 0;

    /* Declaracion de arreglo y matriz iniciando los valores en 0. */
    char nombreAlumnos[numAlum][50] = {" "};
    /* Datos de prueba */
    /* char nombreAlumnos[numAlum][50] = {"Julian", "Chris", "Manuel",
"Ghala", "Joshua"}; */
    float calificaciones[numAlum][unidades] = {0};
    float promedio[numAlum] = {0};

    do
    {
        /* Menu de la practica. */
        printf("\n-----Menu-----");
        printf("\n\t1) Capturar nombres");
        printf("\n\t2) Asignar calificaciones");
        printf("\n\t3) Calcular promedio");
        printf("\n\t4) Mostrar informcion");
        printf("\n\t5) Salir");
        printf("\nSelecciona una opcion:\t");
        scanf("%d", &opc);

        switch (opc)
        {
            case 1:
                AsignarNombres(nombreAlumnos);
                break;
            case 2:
                AsignarCalificaciones(nombreAlumnos, calificaciones,
promedio);
                break;
            case 3:
                CalcularPromedio(nombreAlumnos, calificaciones, promedio);
                break;
            case 4:
                MostrarInformacion(nombreAlumnos, calificaciones,
promedio);
                break;
            case 5:

```

```

        printf("\nSaliendo del programa");
        break;
    default:
        printf("\nOpcion no disponible =(");
        break;
    }
    printf("\n\nPresione cualquier tecla para continua");
    fflush(stdin);
    getchar();
} while (opc != 5);
return 0;
}

int aleatorio(void)
{
    int num;
    num = rand() % 10 + 1;
    return num;
}

void AsignarNombres(char arreglo[][50])
{
    char nombre[50] = "";
    for (int i = 0; i < numAlum; i++)
    {
        printf("\nIngresa el nombre del %d Alumno:", i + 1);
        scanf("%49s", nombre);
        strncpy(arreglo[i], nombre, 50);
    }
}

/* Funcion para rellenar aleatoriamente las calificaciones */
/* void AsignarCalificaciones(float arreglo[numAlum][unidades]){

    printf("\n Alumno   |P1\tP2\tP3\tP4\tP5\t");
    for(int f = 0; f < numAlum; f++){
        printf("\n Alumno %d |", f+1);
        for(int c = 0; c < unidades; c++){
            arreglo[f][c] = aleatorio();
            printf("%.2f  ", arreglo[f][c]);

        }
    }
} */

```

```

void AsignarCalificaciones(char nombreAlumno[numAlum][50], float
calificaciones[numAlum][unidades], float promedio[numAlum])
{
    float cal = 0;
    printf("\n Capturar calificaciones.");
    for (int f = 0; f < numAlum; f++)
    {
        printf("\nAlumno: %s\n", nombreAlumno[f]);
        for (int c = 0; c < unidades; c++)
        {
            printf("Parcial %d:", c + 1);
            scanf("%f", &cal);
            calificaciones[f][c] = cal;
        }
    }
}

void CalcularPromedio(char nombreAlumno[][50], float
calificaciones[numAlum][unidades], float promedio[numAlum])
{
    for (int f = 0; f < numAlum; f++)
    {
        for (int c = 0; c < unidades; c++)
        {
            promedio[f] += calificaciones[f][c];
        }
        promedio[f] = promedio[f] / unidades;
        printf("\n%s:  %0.2f ", nombreAlumno[f], promedio[f]);
    }
}

void MostrarInformacion(char nombreAlumno[][50], float
calificaciones[numAlum][unidades], float promedio[numAlum])
{
    printf("\nAlumno  |P1\tP2\tP3\tP4\tP5\t| Promedio");
    for (int f = 0; f < numAlum; f++)
    {
        printf("\n %s |", nombreAlumno[f]);
        for (int c = 0; c < unidades; c++)
        {
            printf("%.2f\t", calificaciones[f][c]);
        }
        printf(" | %.2f", promedio[f]);
    }
}

```

```
}
```

Compilación y ejecución del código.

```
C:\Users\Okuyt\Desktop\Lenguaje C\Laboratorio\p7>gcc main.c -o main.exe
```

```
C:\Users\Okuyt\Desktop\Lenguaje C\Laboratorio\p7>main.exe
```

```
-----Menu-----
    1) Capturar nombres
    2) Asignar calificaciones
    3) Calcular promedio
    4) Mostrar informcion
    5) Salir
Selecciona una opcion:
```

Menú:

```
-----Menu-----
    1) Capturar nombres
    2) Asignar calificaciones
    3) Calcular promedio
    4) Mostrar informcion
    5) Salir
Selecciona una opcion: |
```

Opción 1, capturar nombres.

```
Selecciona una opcion: 1  
  
Ingresa el nombre del 1 Alumno:Julian  
  
Ingresa el nombre del 2 Alumno:Chris  
  
Ingresa el nombre del 3 Alumno:Manuel  
  
Ingresa el nombre del 4 Alumno:Ghala  
  
Ingresa el nombre del 5 Alumno:Joshua  
  
Presione cualquier tecla para continua
```

En esta parte utilicé la biblioteca string, para utilizar la función strncpy y poder guardar las cadenas de caracteres.

Opción 2, Capturar calificaciones.

```
Capturar calificaciones.  
Alumno: Julian  
Parcial 1:10  
Parcial 2:9  
Parcial 3:9  
Parcial 4:9  
Parcial 5:10
```

```
Alumno: Chris  
Parcial 1:9  
Parcial 2:8  
Parcial 3:8  
Parcial 4:8  
Parcial 5:9
```

```
Alumno: Manuel  
Parcial 1:8  
Parcial 2:8  
Parcial 3:8  
Parcial 4:8  
Parcial 5:8
```

```
Alumno: Ghala  
Parcial 1:7  
Parcial 2:8  
Parcial 3:7  
Parcial 4:8  
Parcial 5:7
```

```
Alumno: Joshua  
Parcial 1:8  
Parcial 2:  
9  
Parcial 3:8  
Parcial 4:7  
Parcial 5:5
```



Opción 3, calcular promedio.

```
Julian:  9.40
Chris:   8.40
Manuel:  8.00
Ghala:   7.40
Joshua:  7.40
```

Opción 4, mostrar información.

Alumno	P1	P2	P3	P4	P5	Promedio
Julian	10.00	9.00	9.00	9.00	10.00	9.40
Chris	9.00	8.00	8.00	8.00	9.00	8.40
Manuel	8.00	8.00	8.00	8.00	8.00	8.00
Ghala	7.00	8.00	7.00	8.00	7.00	7.40
Joshua	8.00	9.00	8.00	7.00	5.00	7.40

Presione cualquier tecla para continua

## Conclusiones

Se logró implementar arreglos y matrices en C para almacenar y procesar información de alumnos y sus calificaciones. Se modularizó el código mediante funciones para capturar datos, calcular promedios y mostrar resultados, mejorando su organización y reutilización. El uso de ciclos permitió el cálculo automático de promedios, optimizando el procesamiento de datos. La interacción con el usuario mediante un menú facilitó la manipulación del programa. Como mejora, se podrían agregar validaciones para evitar calificaciones fuera de rango y mejorar la presentación de la tabla pero es clase de c y no de UX =).

## Bibliografía

[1] "tf." [Online]. Available: <https://la.mathworks.com/help/control/ref/tf.html>.