# UNIVERSIDAD DE LAS AMERICAS (UDLA) FACULTAD DE INGENIERIA Y CIENCIAS APLICADAS CARRERA DE INGENIERIA EN CIBERSEGURIDAD

**ASIGNATURA:** **ISWZ1102 – PROGRAMACIÓN 1**

**DOCENTE:** **EDDY ARMAS**

**ESTUDIANTES:** **MATEO VAZQUEZ E ISRAEL ZURITA**

**FECHA:** **14/10/2025**

## EJERCICIO PROGRAMACIÓN

### OBJETIVOS PROPUESTO DE LA CONSIGNA:

- Reforzar el aprendizaje de utilización de vectores en C.
- Aplicar el uso de vectores y matrices para resolver un problema computacional.
- Reforzar el análisis de problemas para plantear una solución computacional.


## INDICACIONES:

**Descripción del Problema:** Una escuela desea implementar un programa en C que le permita gestionar las calificaciones de un grupo de estudiantes en varias asignaturas. Además de gestionar las calificaciones, el programa debe calcular y mostrar información relevante como el promedio de calificaciones por estudiante y por asignatura, la calificación más alta y baja, y cuántos estudiantes aprobaron cada asignatura.

**Antes de comenzar a programar debe responder a las siguientes preguntas:**

- **¿Cómo organizamos los datos (matrices o vectores)?**
  Para organizar los datos utilizamos matrices ya que nos permiten almacenar las notas de manera ordenada mediante sus índices bidimensionales.
- **¿Qué cálculos son necesarios?**
  En el desarrollo de este programa utilizamos las siguientes operaciones: la suma y promedio de las notas ingresadas por cada estudiante en cada asignatura.
- **¿Cómo manejamos la validación de entradas?**
  La validación de entrada usamos bucles repetitivos los cuales nos permiten verificar si el ingreso de la información es correcto antes de continuar con el programa, los mismos que son: if, do- while y for, que nos permiten establecer un rango especifico en este problema el rango es entro (0 y 10), de tal modo que si la información no es correcta el programa nos indica que los datos ingresados son erróneos.
- **¿Cómo visualizamos los resultados de manera clara?**
  Gracias a la implementación de matrices en el lenguaje C, nos permite ordenar la información y clasificarla de acuerdo a lo sugerido por el programa, en este caso nos permite visualizar los promedios, las máximas notas, las mínimas notas y los estudiantes que aprobaron y reprobaron en cada asignatura.

**Requerimientos funcionales del programa:**

1. El número de estudiantes será 5 y de asignaturas 3.
2. El programa debe pedir las calificaciones de cada estudiante en cada asignatura.
3. Validar que las calificaciones ingresadas estén en el rango de 0 a 10.
4. El programa debe ser capaz de realizar las siguientes tareas:
   - Calcular el promedio de calificaciones para cada estudiante.
   - Calcular el promedio por asignatura.
   - Encontrar la calificación más alta y baja por estudiante y por asignatura.
   - Determinar cuántos estudiantes aprobaron o reprobaron cada asignatura. (Nota aprobatoria $\geq 6$).

**El programa debe mostrar:**

- El promedio de calificaciones por estudiante.
- El promedio de calificaciones por asignatura.
- La calificación más alta y baja por estudiante y por asignatura.
- El número de estudiantes aprobados y reprobados por asignatura.

## FORMA DE TRABAJO:

El trabajo se desarrollará en grupos de máximo 2 integrantes.

## ESPECIFICACIONES DE ENTREGA:

Los estudiantes deberán elaborar un informe con la siguiente estructura:

1. **Introducción**
    - **Explica el propósito del programa que desarrollaste.**
      El propósito del programa es calcular los promedios que tiene cada estudiante en cada asignatura de forma ordenada mediante el uso de matrices, las cuales nos indican como ha sido el rendimiento académico de cada estudiante, siempre y cuando los datos ingresados se alineen a los requerimientos solicitados por el usuario.
    - **¿Qué problema resuelve y por qué es importante?**
      El problema que resuelve es que los docentes pueden registrar de forma más rápida y fácil las calificaciones de sus estudiantes mediante el uso de matrices en este programa, ya que su uso conlleva a optimizar el tiempo al calcular los promedios de forma automatizada y no de forma manual lo que lo hace más eficiente.

2. **Implementación del Código**
    - Imágenes del código desarrollado.

```
Welcome to GDB Online.
  GDB online is an online compiler and debugger tool for C, C++, Python, PHP, Ruby,
  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, COBOL, HTML, CSS, JS
  Code, Compile, Run and Debug online from anywhere in world.

*******************************************************************************/
#include <stdio.h>

#define ESTUDIANTES 5
#define ASIGNATURAS 3

int main() {
 float calificaciones[ESTUDIANTES][ASIGNATURAS]// crea la matriz de 5 filas y 5 colu
 int i, j;
 float sumaEst, sumaAsig;
 float promedioEst[ESTUDIANTES], promedioAsig[ASIGNATURAS];
 float maxEst[ESTUDIANTES], minEst[ESTUDIANTES];
 float maxAsig[ASIGNATURAS], minAsig[ASIGNATURAS];
 int aprobados[ASIGNATURAS] = {0};
 int reprobados[ASIGNATURAS] = {0};
```

```
// Ingreso de calificaciones con validación
for(i = 0; i < ESTUDIANTES; i++) {
printf("Ingresar calificaciones para estudiante %d:\n", i+1);
for(j = 0; j < ASIGNATURAS; j++) {
do {
printf(" Asignatura %d (0-10): ", j+1);
scanf("%f", &calificaciones[i][j]);
if(calificaciones[i][j] < 0 || calificaciones[i][j] > 10) {
printf(" Nota inválida, debe estar entre 0 y 10.\n");
}
} while(calificaciones[i][j] < 0 || calificaciones[i][j] > 10);
}
}
```

```
// Inicializar valores para max/min
for(i = 0; i < ESTUDIANTES; i++) {
maxEst[i] = calificaciones[i][0];
minEst[i] = calificaciones[i][0];
}
for(j = 0; j < ASIGNATURAS; j++) {
maxAsig[j] = calificaciones[0][j];
minAsig[j] = calificaciones[0][j];
}

// Cálculos
// Promedio, max y min por estudiante
for(i = 0; i < ESTUDIANTES; i++) {
sumaEst = 0;
for(j = 0; j < ASIGNATURAS; j++) {
float nota = calificaciones[i][j];
sumaEst += nota;
if(nota > maxEst[i]) maxEst[i] = nota;
if(nota < minEst[i]) minEst[i] = nota;
}
promedioEst[i] = sumaEst / ASIGNATURAS;
}
```

```
// Promedio, max, min y conteo aprobados/reprobados por asignatura
for(j = 0; j < ASIGNATURAS; j++) {
sumaAsig = 0;
for(i = 0; i < ESTUDIANTES; i++) {
float nota = calificaciones[i][j];
sumaAsig += nota;
if(nota > maxAsig[j]) maxAsig[j] = nota;
if(nota < minAsig[j]) minAsig[j] = nota;
if(nota >= 6) aprobados[j]++;
else reprobados[j]++;
}
promedioAsig[j] = sumaAsig / ESTUDIANTES;
}
```

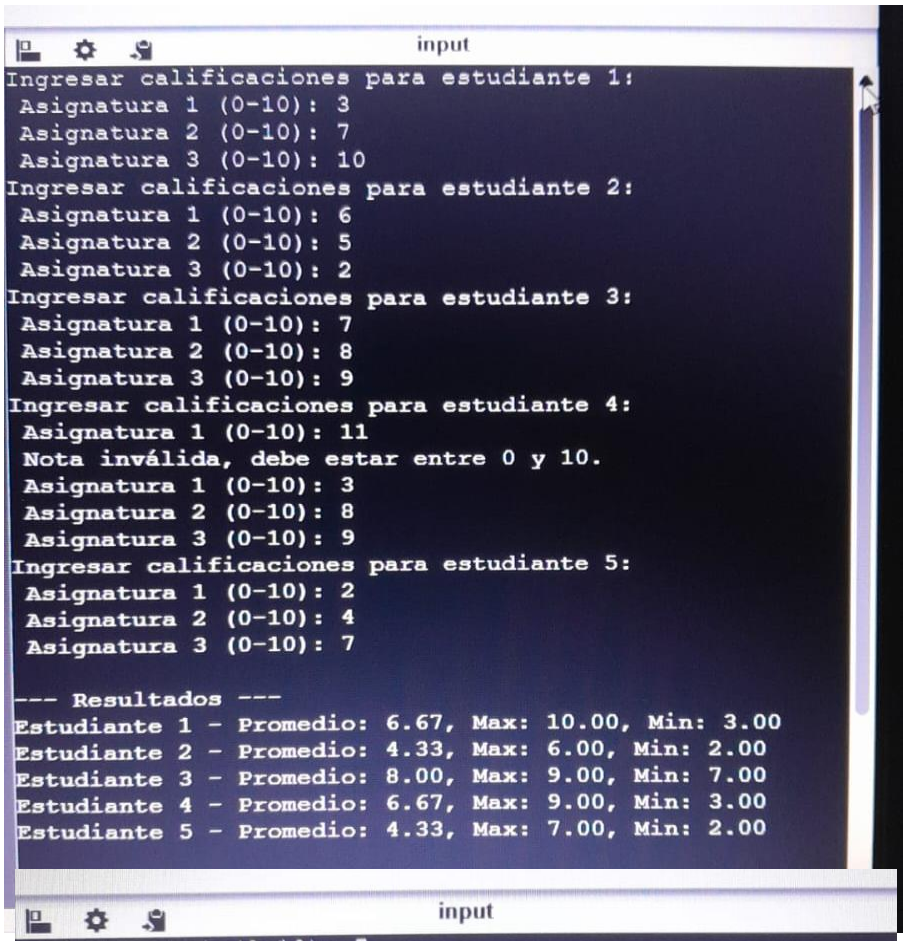
```

// Mostrar resultados
printf("\n--- Resultados ---\n");

for(i = 0; i < ESTUDIANTES; i++) {
printf("Estudiante %d - Promedio: %.2f, Max: %.2f, Min: %.2f\n", i+1, promedioEst[i], maxEst[i], minEst[i]);
}

printf("\nPor Asignatura:\n");
for(j = 0; j < ASIGNATURAS; j++) {
printf("Asignatura %d - Promedio: %.2f, Max: %.2f, Min: %.2f, Aprobados: %d, Reprobados: %d\n",
j+1, promedioAsig[j], maxAsig[j], minAsig[j], aprobados[j], reprobados[j]);
}

return 0;
}
```

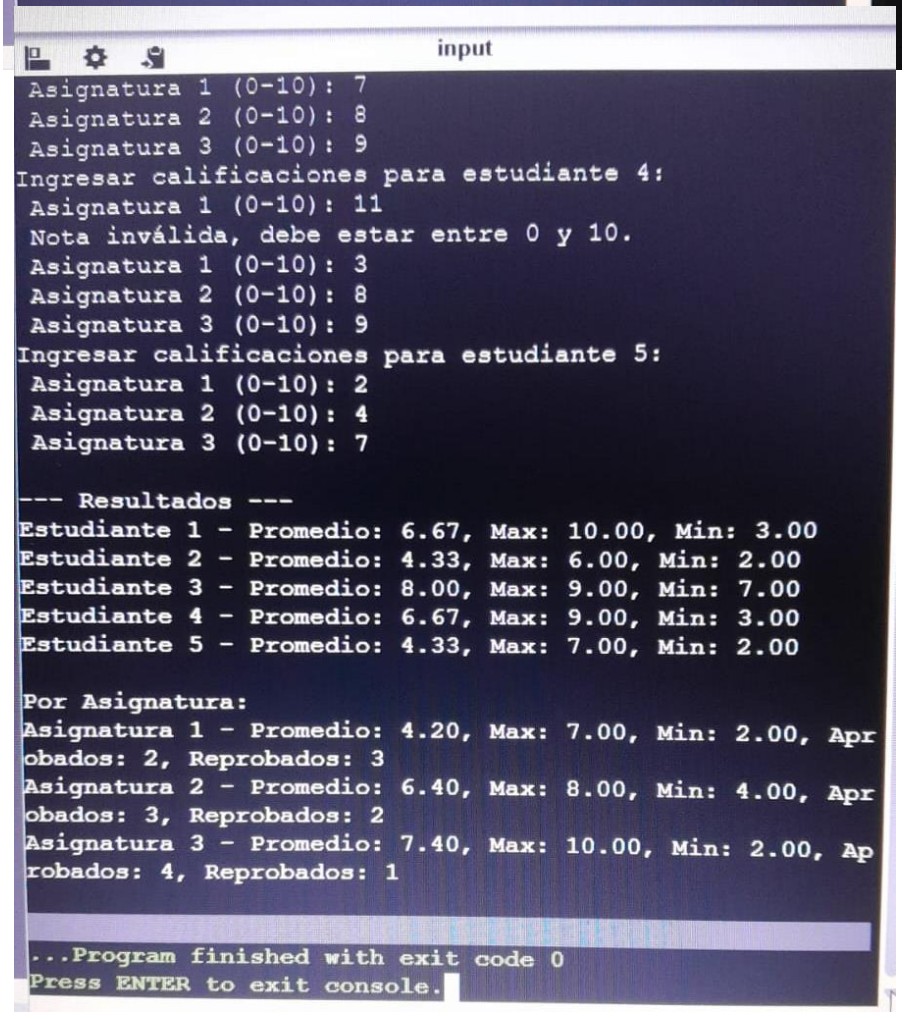
- **Enlace de Github con el código del programa.**

https://github.com/JeremyVasquez/taller_N2_JeremyVasquez_IsraelZurita-.git

## 3. Validación y Pruebas

- Imágenes de la ejecución del código.


input
Ingresar calificaciones para estudiante 1:
Asignatura 1 (0-10): 3
Asignatura 2 (0-10): 7
Asignatura 3 (0-10): 10
Ingresar calificaciones para estudiante 2:
Asignatura 1 (0-10): 6
Asignatura 2 (0-10): 5
Asignatura 3 (0-10): 2
Ingresar calificaciones para estudiante 3:
Asignatura 1 (0-10): 7
Asignatura 2 (0-10): 8
Asignatura 3 (0-10): 9
Ingresar calificaciones para estudiante 4:
Asignatura 1 (0-10): 11
Nota inválida, debe estar entre 0 y 10.
Asignatura 1 (0-10): 3
Asignatura 2 (0-10): 8
Asignatura 3 (0-10): 9
Ingresar calificaciones para estudiante 5:
Asignatura 1 (0-10): 2
Asignatura 2 (0-10): 4
Asignatura 3 (0-10): 7
--- Resultados ---
Estudiante 1 - Promedio: 6.67, Max: 10.00, Min: 3.00
Estudiante 2 - Promedio: 4.33, Max: 6.00, Min: 2.00
Estudiante 3 - Promedio: 8.00, Max: 9.00, Min: 7.00
Estudiante 4 - Promedio: 6.67, Max: 9.00, Min: 3.00
Estudiante 5 - Promedio: 4.33, Max: 7.00, Min: 2.00


input
Asignatura 1 (0-10): 7
Asignatura 2 (0-10): 8
Asignatura 3 (0-10): 9
Ingresar calificaciones para estudiante 4:
Asignatura 1 (0-10): 11
Nota inválida, debe estar entre 0 y 10.
Asignatura 1 (0-10): 3
Asignatura 2 (0-10): 8
Asignatura 3 (0-10): 9
Ingresar calificaciones para estudiante 5:
Asignatura 1 (0-10): 2
Asignatura 2 (0-10): 4
Asignatura 3 (0-10): 7
--- Resultados ---
Estudiante 1 - Promedio: 6.67, Max: 10.00, Min: 3.00
Estudiante 2 - Promedio: 4.33, Max: 6.00, Min: 2.00
Estudiante 3 - Promedio: 8.00, Max: 9.00, Min: 7.00
Estudiante 4 - Promedio: 6.67, Max: 9.00, Min: 3.00
Estudiante 5 - Promedio: 4.33, Max: 7.00, Min: 2.00
Por Asignatura:
Asignatura 1 - Promedio: 4.20, Max: 7.00, Min: 2.00, Aprobados: 2, Reprobados: 3
Asignatura 2 - Promedio: 6.40, Max: 8.00, Min: 4.00, Aprobados: 3, Reprobados: 2
Asignatura 3 - Promedio: 7.40, Max: 10.00, Min: 2.00, Aprobados: 4, Reprobados: 1
...Program finished with exit code 0
Press ENTER to exit console.

## 4. Conclusiones

- **¿Qué aprendiste al desarrollar este programa?**

  Aprendimos a usar matrices para manejar varios datos (notas de estudiantes y materias). Se practicó cómo validar datos para evitar errores al ingresar información. También entendimos cómo recorrer filas y columnas para sacar promedios, máximos, mínimos y contar aprobados/reprobados.

- **¿Cuáles fueron los principales retos que enfrentaste antes de comenzar a programar?**

  Antes de comenzar el programa los retos que se nos presentaron fueron que al principio se dificulto entender cómo almacenar la información debido a que el uso de una matriz bidimensional (estudiantes y asignaturas) no es intuitivo al inicio, requiere visualizar que cada fila representa un estudiante y cada columna una asignatura y ese salto conceptual cuesta, porque implica pensar en coordenadas y no solo en valores sueltos. Además, el uso de los bucles anidados, es decir, las condiciones (if, do while), y las validaciones (“nota entre 0 y 10”) ya que no basta con saber la sintaxis, hay que entender cuándo y por qué usar cada estructura. Aquí entra en juego la lógica algorítmica, la cual se entrena con la práctica.