



Menu – Menu Anidado

1. DATOS INFORMATIVOS

Carrera: Electronica y Automatizacion

Asignatura: Fundamento de Programación

Tema del taller: Menu-Menu Anidado

Docente: Ing. Jenny Ruiz

Integrantes: Mendoza Cortez Christopher Alexander

Fecha: 26/11/2025 Paralelo: 29583

2. DESARROLLO

Desarrollo de Menú Principal (MP) con 1- Expresiones y 2 Ecuacion de 2do Grado, el sub menu con 6 expresiones. en cada cada uno debe haber regresar el Menu Principal.

Codigo

```
#include <stdio.h>
#include <math.h>

// Submenu para las expresiones

void mostrarExpresiones() {
    int opcion;
    do {
        printf("\n-- SUBMENU EXPRESIONES --\n");
        printf("1. Ejercicio 1\n");
        printf("2. Ejercicio 2\n");
        printf("3. Ejercicio 3\n");
        printf("4. Ejercicio 4\n");
```

```

printf("5. Ejercicio 5\n");

printf("6. Ejercicio 6\n");

printf("7. Regresar al Menu Principal\n");

printf("Seleccione una opcion:");

scanf("%d", &opcion);

switch(opcion){

    case 1: {

        int i=4,j=2,k=8;

        int resultado = (3*j + 8/k) != i*k - j;

        printf("\nEjercicio 1:\n(3*j + 8/k) != i*k - j\n");

        printf("Sustitucion: (3*%d + 8/%d) != %d*%d - %d\n",j,k,i,k,j);

        printf("Calculo: (%d + %d) != (%d - %d)\n", 3*j, 8/k, i*k, j);

        printf("Resultado: %d\n", resultado);

        break;

    }

    case 2: {

        int m=6,n=3,p=9;

        int resultado = m*(n+p)/2>= p-n;

        printf("\nEjercicio 2:\nm*(n+p)/2>= p-n\n");

        printf("Sustitucion: %d*(%d+%d)/2>= %d-%d\n", m,n,p,p,n);

        printf("Calculo: (%d)/2>= %d\n", m*(n+p), p-n);

        printf("Resultado: %d\n", resultado);

        break;

    }

}

```

case 3: {

```

int a=5,b=2,c=10;

int resultado = (a + b*c) < (c/b + a);

printf("\nEjercicio 3:\n(a + b*c) < (c/b + a)\n");

printf("Sustitucion: (%d + %d * %d) < (%d / %d + %d)\n", a,b,c,c,b,a);

printf("Calculo: (%d + %d) < (%d + %d)\n", a,b*c,c/b,a);

printf("Resultado: %d\n", resultado);

break;

}

```

case 4: {

```

int x=14,y=3;

int resultado = x/(y-1)+3*y == x-y;

printf("\nEjercicio 4:\nx/(y-1)+3*y == x-y\n");

printf("Sustitucion: %d / (%d-1) + 3 * %d == %d - %d\n", x,y,y,x,y);

printf("Calculo: (%d) + (%d) == %d\n", x/(y-1), 3*y, x-y);

printf("Resultado: %d\n", resultado);

break;

}

```

case 5: {

```

int u=12,v=4;

int resultado = (u*v-4) <= (v+u/2);

printf("\nEjercicio 5:\n(u*v-4) <= (v+u/2)\n");

printf("Sustitucion: (%d * %d - 4) <= (%d + %d / 2)\n", u,v,v,u);

printf("Calculo: (%d - 4) <= (%d + %d)\n", u*v,v,u/2);

printf("Resultado: %d\n", resultado);

```

```

break;

}

case 6: {

    int q=7,r=5;

    int resultado = (q+r*2) !=(q*r-10);

    printf("\nEjercicio 6:\n(q+r*2) !=(q*r-10)\n");

    printf("Sustitucion: (%d+%d*2) != (%d*%d-10)\n", q,r,q,r);

    printf("Calculo: (%d+%d) != (%d-10)\n", q,r*2,q*r);

    printf("Resultado: %d\n", resultado);

    break;

}

case 7:

printf("Regresando al Menu Principal...\n");

break;

default:

printf("Opcion invalida.\n");

}

} while(opcion != 7);

}

// Resolver ecuación de segundo grado

void resolverEcuacion() {

double a,b,c;

printf("\n--- Resolver Ecuacion de Segundo Grado ---\n");

printf("Ingrese coeficiente a: ");

```



```
scanf("%lf", &a);

printf("Ingrese coeficiente b:");

scanf("%lf", &b);

printf("Ingrese coeficiente c:");

scanf("%lf", &c);

if(a == 0) {

    printf("Error: a no puede ser 0 (no es ecuacion cuadratica). \n");

    return;

}

double discriminante = b*b - 4*a*c;

if(discriminante < 0) {

    printf("La ecuacion no tiene soluciones reales. \n");

} else {

    double x1 = (-b + sqrt(discriminante)) / (2*a);

    double x2 = (-b - sqrt(discriminante)) / (2*a);

    printf("Soluciones: x1= %.2lf, x2 = %.2lf\n", x1, x2);

}

}

int main(){

int opcion;

do {

printf("\n==== MENU PRINCIPAL====\n");


```



```
printf("1. Expresiones\n");

printf("2. Ecuacion de Segundo Grado\n");

printf("3. Salir\n");

printf("Seleccione una opcion: ");

scanf("%d", &opcion);

switch(opcion){

    case 1:

        mostrarExpresiones();

        break;

    case 2:

        resolverEcuacion();

        break;

    case 3:

        printf("Saliendo del programa...\n");

        break;

    default:

        printf("Opcion invalida.\n");

}

} while(opcion != 3);

return 0;
}
```

Prueba de Escritorio

```

C:\Users\Katy Mendoza\Desktop > + 
--- MENU PRINCIPAL ---
1. Expresiones
2. Ecuacion de Segundo Grado
3. Salir
Seleccione una opcion:
1

--- SUBMENU EXPRESIONES ---
1. Ejercicio 1
2. Ejercicio 2
3. Ejercicio 3
4. Ejercicio 4
5. Ejercicio 5
6. Ejercicio 6
7. Regresar al Menu Principal
Seleccione una opcion: 1

Ejercicio 1:
 $(3*j + 8/k) \neq i*k - j$ 
Sustitucion:  $(3*2 + 8/8) \neq 4*8 - 2$ 
Calculo:  $(6 + 1) \neq (32 - 2)$ 
Resultado: 1

--- SUBMENU EXPRESIONES ---
1. Ejercicio 1
2. Ejercicio 2
3. Ejercicio 3
4. Ejercicio 4
5. Ejercicio 5
6. Ejercicio 6
7. Regresar al Menu Principal
Seleccione una opcion:
1

Ejercicio 2:
 $m*(n+p)/2 \geq p-n$ 
Sustitucion:  $6*(3+9)/2 \geq 9-3$ 
Calculo:  $(72)/2 \geq 6$ 
Resultado: 1

--- SUBMENU EXPRESIONES ---
1. Ejercicio 1
2. Ejercicio 2
3. Ejercicio 3
4. Ejercicio 4
5. Ejercicio 5
6. Ejercicio 6
7. Regresar al Menu Principal
Seleccione una opcion: 3

Ejercicio 3:
 $(a + b*c) < (c/b + a)$ 
Sustitucion:  $(5 + 2*10) < (10/2 + 5)$ 
Calculo:  $(5 + 20) < (5 + 5)$ 
Resultado: 0

C:\Users\Katy Mendoza\Desktop > + 
Resultado: 0

--- SUBMENU EXPRESIONES ---
1. Ejercicio 1
2. Ejercicio 2
3. Ejercicio 3
4. Ejercicio 4
5. Ejercicio 5
6. Ejercicio 6
7. Regresar al Menu Principal
Seleccione una opcion: 7
Regresando al Menu Principal...

--- MENU PRINCIPAL ---
1. Expresiones
2. Ecuacion de Segundo Grado
3. Salir
Seleccione una opcion: 2

--- Resolver Ecuacion de Segundo Grado ---
Ingrese coeficiente a: 2
Ingrese coeficiente b: 5
Ingrese coeficiente c: -3
Soluciones: x1 = 0.50, x2 = -3.00

--- MENU PRINCIPAL ---
1. Expresiones
2. Ecuacion de Segundo Grado
3. Salir
Seleccione una opcion: 3
Saliendo del programa...

Process returned 0 (0x0)  execution time : 90.055 s
Press any key to continue.
|

```

3. CONCLUSIONES

- Aprender a evaluar expresiones en C ayuda a entender cómo funciona la lógica y el orden de operaciones en programación. Esto es clave para escribir código correcto y evitar errores.
- Resolver ecuaciones de segundo grado en C demuestra cómo aplicar matemáticas en la práctica. Combinar teoría y programación permite crear soluciones útiles y automatizadas.

4. RECOMENDACIONES

- Antes de programar expresiones en C, repasa el **orden de operadores** (aritméticos, relacionales y lógicos). Esto evita errores y asegura que el resultado sea el esperado.
- Cuando trabajes con ecuaciones de segundo grado, **valida los datos** (por ejemplo, que $a \neq 0$ y que el discriminante no sea negativo) para evitar cálculos incorrectos o errores en tiempo de ejecución.

5. REFERENCIAS