

Uso de menús con figuras y series en C

(Programador Junior en Code::Blocks)

1. DATOS INFORMATIVOS

Carrera: Electronica y Automatizacion

Asignatura: Fundamentos de Programacion

Tema del taller: Uso de menús con figuras y series en C

Docente: Ing. Jenny Ruiz

Integrantes: Mendoza Cortez Christopher Alexander

Fecha: 28/11/2025 Paralelo: 29583

2. DESARROLLO

En este taller vas a trabajar con un programa en lenguaje C que utiliza un menú para elegir entre dos opciones principales:

- 1) Dibujar la figura de un cuadrado HUECO con el número 0 en la pantalla.
- 2) Calcular el valor de una serie numérica una vez ingresado un valor desde teclado

El objetivo es que practiques el uso de estructuras de control como do–while, switch, for y if, así como la organización de un programa con menú en Code::Blocks

Paso 1: Crear un nuevo proyecto en Code::Blocks

Paso 2: Copiar el código base del menu

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

int main() {
    int op;
    int fil, i, j;
    int n;
    double res;

    do {
        printf("\n==== MENU PRINCIPAL ====\n");
        printf("1. Dibujar la figura de un cuadrado HUECO con el número 0 en la pantalla.\n");
        printf("2. Calcular el valor de una serie numérica una vez ingresado un valor desde teclado\n");
        printf("3. Salir\n");
        printf("Ingresa tu opción: ");
        scanf("%d", &op);

        switch (op) {
            case 1:
                // Código para dibujar cuadrado hueco
                break;
            case 2:
                // Código para calcular serie numérica
                break;
            case 3:
                printf("Saliendo...\n");
                break;
            default:
                printf("Opción no válida. Inténtalo de nuevo.\n");
        }
    } while (op != 3);
}
```

```

printf("1) Figura con 1's (cuadrado)\n");
printf("2) Serie numerica\n");
printf("3) Salir\n");
printf("Elige una opcion: ");
scanf("%d", &op);

switch (op) {
    case 1:
        // Opcion 1: dibujar un cuadrado hueco de 1's
        do {
            printf("\nIngresa el numero de filas (mayor que 0): ");
            scanf("%d", &fil);
        } while (fil <= 0);

        for (i = 1; i <= fil; i++) {
            for (j = 1; j <= fil; j++) {
                // Imprime 1 en los bordes, espacio en el interior
                if (i == 1 || i == fil || j == 1 || j == fil) {
                    printf("1 ");
                } else {
                    printf("  "); // dos espacios para mantener alineación
                }
            }
            printf("\n");
        }
        break;

    case 2:
        // Opcion 2: serie 1^1 + 2^2 + 3^3 + ... + n^n
        do {
            printf("\nIngresa un valor para n (mayor que 0): ");
            scanf("%d", &n);
        } while (n <= 0);

        res = 0;
        printf("\nSerie: ");
        for (i = 1; i <= n; i++) {
            res = res + pow(i, i);
            printf("%d^%d ", i, i);
            if (i < n) {
                printf(" + ");
            }
        }
        printf("\nResultado de la serie = %.0f\n", res);
        break;
}

```

```

case 3:
    printf("\nSaliendo del programa...\n");
    break;

default:
    printf("\nOpcion no valida. Intenta de nuevo.\n");
}
} while (op != 3);

return 0;
}

```

Paso 3: Actividad A – Modificar la figura

Actualmente, la opción 1 dibuja un cuadrado hueco de 1's. Tu tarea es modificar el código para que, además de los bordes, se dibuje la diagonal principal del cuadrado con 1's.

Sugerencias para la modificación:

- Identifica el bloque if dentro del doble for donde se decide cuándo imprimir "1" o espacios.
- Recuerda que en la diagonal principal se cumple que $i == j$.
- Agrega esa condición al if para que también imprima "1" cuando $i == j$.

Prueba la nueva figura con diferentes valores de fil (por ejemplo, 3, 4 y 5) y verifica que se vea el cuadrado con la diagonal marcada.

Código Modificado

```

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

int main() {
    int op;
    int fil, i, j;
    int n;
    double res;

    do {
        printf("\n==== MENU PRINCIPAL ====\n");
        printf("1) Figura con 0's (cuadrado)\n");
        printf("2) Serie numerica\n");
        printf("3) Salir\n");
        printf("Elige una opcion: ");
        scanf("%d", &op);

```

```

switch (op) {
case 1:
    // Opcion 1: dibujar un cuadrado hueco de 0's con diagonal principal
    do {
        printf("\nIngresa el numero de filas (mayor que 0): ");
        scanf("%d", &fil);
    } while (fil <= 0);

    for (i = 1; i <= fil; i++) {
        for (j = 1; j <= fil; j++) {
            // Imprime 0 en los bordes o en la diagonal principal
            if (i == 1 || i == fil || j == 1 || j == fil || i == j) {
                printf("0 ");
            } else {
                printf("  "); // dos espacios para mantener alineación
            }
        }
        printf("\n");
    }
    break;

case 2:
    // Opcion 2: serie 1^1 + 2^2 + 3^3 + ... + n^n (original)
    do {
        printf("\nIngresa un valor para n (mayor que 0): ");
        scanf("%d", &n);
    } while (n <= 0);

    res = 0;
    printf("\nSerie: ");
    for (i = 1; i <= n; i++) {
        res = res + pow(i, i);
        printf("%d^%d ", i, i);
        if (i < n) {
            printf(" + ");
        }
    }
    printf("\nResultado de la serie = %.0f\n", res);
    break;

case 3:
    printf("\nSaliendo del programa...\n");
    break;

default:
    printf("\nOpcion no valida. Intenta de nuevo.\n");
}

```

```

        }
    } while (op != 3);

    return 0;
}

```

Paso 4: Actividad B – Modificar la serie

La opción 2 calcula la serie $1^1 + 2^2 + 3^3 + \dots + n^n$. Ahora vas a cambiar la serie para que sea:

$$S = 1^2 + 2^2 + 3^2 + \dots + n^2$$

Para hacerlo, modifica dentro del case 2:

- Cambia la operación de acumulación por $res = res + i * i;$
- Ajusta el printf dentro del for para que muestre i^2 en lugar de i^i .

Prueba la nueva versión con distintos valores de n (por ejemplo, 3, 4 y 5) y comprueba mentalmente o con una calculadora que el resultado sea correcto.

Código Modificado

```

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

int main() {
    int op;
    int fil, i, j;
    int n;
    double res;

    do {
        printf("\n==== MENU PRINCIPAL ====\n");
        printf("1) Figura con 0's (cuadrado)\n");
        printf("2) Serie numérica\n");
        printf("3) Salir\n");
        printf("Elige una opción: ");
        scanf("%d", &op);

        switch (op) {
            case 1:
                // Opción 1: dibujara un cuadrado hueco de 0's con diagonal principal
                do {
                    printf("\nIngresa el número de filas (mayor que 0): ");
                    scanf("%d", &fil);
                } while (fil <= 0);

                for (i = 1; i <= fil; i++) {

```

```

for (j = 1; j <= fil; j++) {
    if (i == 1 || i == fil || j == 1 || j == fil || i == j) {
        printf("0 ");
    } else {
        printf("  "); // dos espacios para mantener alineación
    }
    printf("\n");
}
break;

case 2:
// Opcion 2: serie 1^2 + 2^2 + 3^2 + ... + n^2
do {
    printf("\nIngresa un valor para n (mayor que 0): ");
    scanf("%d", &n);
} while (n <= 0);

res = 0;
printf("\nSerie: ");
for (i = 1; i <= n; i++) {
    res = res + i * i;
    printf("%d^2", i);
    if (i < n) {
        printf(" + ");
    }
}
printf("\nResultado de la serie = %.0f\n", res);
break;

case 3:
printf("\nSaliendo del programa...\n");
break;

default:
printf("\nOpcion no valida. Intenta de nuevo.\n");
}
} while (op != 3);

return 0;
}

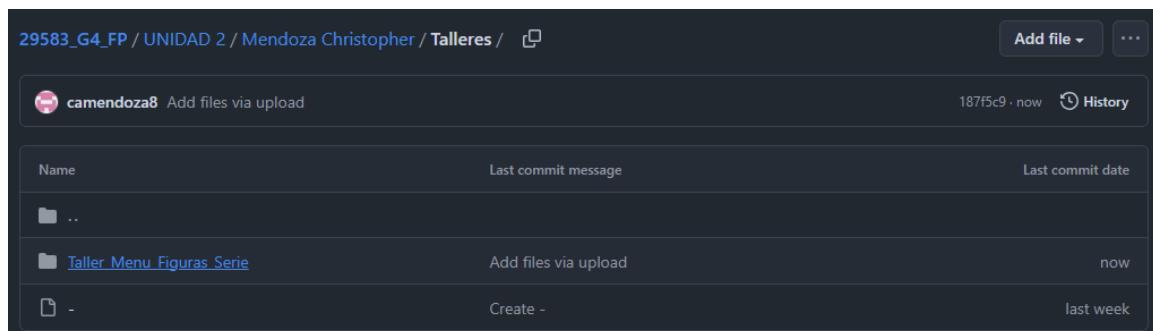
```

Paso 5: Pruebas y evidencias a entregar

Al finalizar el taller, deberás entregar:

El archivo del proyecto de Code::Blocks

La carpeta estara subida en GITHUB en la carpeta personal (se adjunta evidencia)



The screenshot shows a GitHub repository interface. At the top, it displays the path: 29583_G4_FP / UNIDAD 2 / Mendoza Christopher / Talleres /. On the right side, there are buttons for 'Add file' and '...'. Below this, a commit from user 'camendoza8' is shown, with the message 'Add files via upload' and a timestamp of '187f5c9 · now'. A 'History' link is also present. The main area lists three files: a folder named '..', a folder named 'Taller Menu Figuras Serie' with a commit message 'Add files via upload' and timestamp 'now', and a file named '-' with a commit message 'Create -' and timestamp 'last week'.

Capturas de pantalla donde se vea:

El menú principal.

Al presionar f9 ejecutamos el programa y nos muestra por pantalla el menu principal.

```
== MENU PRINCIPAL ==
1) Figura con 0's (cuadrado)
2) Serie numerica
3) Salir
Elige una opcion:
```

La ejecución de la opción 1 con tu figura modificada.

Al presionar 1 nos pedira un numero de filas para poder formar el cuadrado y su diagonal (los numeros deben ser >=3 par que se pueda visibilizar la diagonal del cuadrado).

```
== MENU PRINCIPAL ==
1) Figura con 0's (cuadrado)
2) Serie numerica
3) Salir
Elige una opcion: 1

Ingresa el numero de filas (mayor que 0): 5
0 0 0 0 0
0 0      0
0      0 0
0      0 0
0 0 0 0 0
```

La ejecución de la opción 2 con la nueva serie.

All presionar 2 no s pide un numero para dar valor a n y asi comenzara realizar la serie numerica (los numeros deben ser mayor que 0).

```
== MENU PRINCIPAL ==
1) Figura con 0's (cuadrado)
2) Serie numerica
3) Salir
Elige una opcion: 2

Ingresa un valor para n (mayor que 0): 5

Serie: 1^1 + 2^2 + 3^3 + 4^4 + 5^5
Resultado de la serie = 3413
```

Una breve nota (3-5 líneas) indicando qué fue lo más fácil y lo más difícil del taller

Lo más fácil del taller fue comprender la lógica básica de los bucles y las condiciones para dibujar figuras, ya que se aplican reglas simples y visibles en la salida.

Lo más difícil fue manejar la alineación y las modificaciones específicas (como agregar diagonales o cambiar la serie), porque requieren atención a la sintaxis y a la estructura del código para evitar errores.

Con el código compilado de Codeblocks pasa a GDBOnline para compartir el link

<https://onlinegdb.com/8maFqJRP2>

3. CONCLUSIONES

El taller permitió reforzar conceptos básicos de programación en C, como el uso de bucles, condiciones y operaciones matemáticas, aplicándolos en ejercicios prácticos.

Las modificaciones solicitadas (figuras y series) demostraron la importancia de comprender la lógica detrás del código para adaptarlo a nuevos requerimientos.

4. RECOMENDACIONES

Practicar más ejercicios que combinen estructuras de control y cálculos matemáticos para afianzar la lógica y la sintaxis del lenguaje C.

5. REFERENCIAS