

1. DATOS INFORMATIVOS

Carrera: Electronica y Automatization

Asignatura: Fundamentos de la programacion

Tema del taller: Programador junior que utiliza Code::Blocks

Docente: Ing. Jenny Ruiz

Integrantes: Jefferson Caisaguano

Fecha: 28/11/2025 Paralelo: 29583

2. DESARROLLO

1. Introducción

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.

2. Objetivos de aprendizaje

- Comprender la estructura básica de un menú en C usando do-while y switch.
- Identificar y modificar bucles anidados para dibujar figuras en consola.
- Implementar y ajustar series numéricas usando ciclos for y acumuladores.
- Compilar, ejecutar y probar programas en Code::Blocks de forma autónoma.

3. Requisitos previos

1. Tener instalado Code::Blocks (versión con compilador incluido).

2. Saber crear, compilar y ejecutar un programa sencillo en C (por ejemplo, "Hola mundo").

3. Conocer los tipos básicos de datos (int, float) y el uso de printf/scanf.

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

Sigue estos pasos en Code::Blocks:

4. Abre Code::Blocks.
5. Ve al menú File → New → Project...
6. Selecciona "Console application" y haz clic en Go.
7. Elige el lenguaje C cuando se te pregunte.
8. Asigna un nombre al proyecto, por ejemplo: Taller_Menu_Figuras_Serie.
9. Elige la carpeta donde se guardará el proyecto y termina el asistente.
10. Abre el archivo fuente principal (normalmente main.c).

5. Paso 2: Copiar el código base del menú

En el archivo main.c, borra el contenido que tenga y copia el siguiente código. Lee los comentarios con atención, ya que te indican qué hace cada parte.

```
#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 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(" ");

}

}

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;

}
```

Guarda el archivo (Ctrl + S), compila (F9 o Build → Build) y ejecuta (Ctrl + F10 o Build → Run). Prueba cada opción del menú con diferentes valores.

6. 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.

7. 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.

8. Paso 5: Pruebas y evidencias a entregar

Al finalizar el taller, deberás entregar (según indique tu docente):

- El archivo del proyecto de Code::Blocks (carpeta completa o al menos el main.c).
- Capturas de pantalla donde se vea:
 - El menú principal.
 - La ejecución de la opción 1 con tu figura modificada.
 - La ejecución de la opción 2 con la nueva serie.
- Una breve nota (3–5 líneas) indicando qué fue lo más fácil y lo más difícil del taller.

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

Resolucion

```
*main.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  int main() {
6      int op;
7      int fil, i, j;
8      int n;
9      double res;
10
11      do {
12          printf("\n=== MENU PRINCIPAL ===\n");
13          printf("1) Figura con 1's (cuadrado con diagonal)\n");
14          printf("2) Serie numerica (1^2 + 2^2 + ... + n^2)\n");
15          printf("3) Salir\n");
16          printf("Elige una opcion: ");
17          scanf("%d", &op);
18
19          switch (op) {
20              case 1:
21
22                  do {
23                      printf("\nIngresa el numero de filas (mayor que 0): ");
24                      scanf("%d", &fil);
25                  } while (fil <= 0);
26
27                  for (i = 1; i <= fil; i++) {
28                      for (j = 1; j <= fil; j++) {
29
30                          if (i == 1 || i == fil || j == 1 || j == fil || i == j) {
31                              printf("1 ");
32                          } else {
33                              printf(" ");
34                          }
35                      }
36                      printf("\n");
37                  }
38                  break;
39
40              case 2:
41
42                  do {
43                      printf("\nIngresa un valor para n (mayor que 0): ");
44                      scanf("%d", &n);
45                  } while (n <= 0);
46
47                  res = 0;
48                  printf("\nSerie: ");
49                  for (i = 1; i <= n; i++) {
```



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

```
res = res + (double)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;
}
```

```
C:\Users\G300\Downloads\Tal X + v

Ingresa el numero de filas (mayor que 0): 5
1 1 1 1 1
1 1     1
1  1   1
1     1 1
1 1 1 1 1

=== MENU PRINCIPAL ===
1) Figura con 1's (cuadrado con diagonal)
2) Serie numerica (1^2 + 2^2 + ... + n^2)
3) Salir
Elige una opcion: |
```

```
C:\Users\G300\Downloads\Tal X + v
=== MENU PRINCIPAL ===
1) Figura con 1's (cuadrado con diagonal)
2) Serie numerica (1^2 + 2^2 + ... + n^2)
3) Salir
Elige una opcion: 2

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

Serie: 1^2 + 2^2 + 3^2 + 4^2 + 5^2
Resultado de la serie = 55
```

Nota del taller

Lo fácil del taller fue **comprender y adaptar la lógica**.

La parte más difícil fue **lograr la alineación del cuadrado hueco y la diagonal**, ya que se requirió un ajuste complicado en los espacios (" ") dentro del bloque `else` para que la figura no se viera distorsionada en la consola.

<https://onlinegdb.com/gXqiwbsQC>

3. CONCLUSIONES

El ejercicio permitió reforzar el uso de estructuras como `switch`, `for` y `do-while`, fundamentales para la creación de menús interactivos y validación de datos.

4. RECOMENDACIONES

Realizar comentarios claros en cada bloque para facilitar la comprensión y mantenimiento del código.

5. REFERENCIAS

Joyanes Aguilar, L., & Zahonero Martínez, I. (2008). Programación en C: metodología, algoritmos y estructura de datos (2ª ed.). McGraw-Hill Interamericana.