



**UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO**

INGENIERIA EN SOFTWARE Y TECNOLOGIAS EMERGENTES

GRUPO: 932

MATERIA: Lenguaje C

MAESTRA: Yulith Vanessa Altamirano Flores

TITULO:
Práctica 4. Funciones con retorno y sin retorno.

ALUMNA: Teresa Rivas Gómez

MATRICULA: 372565

Práctica 4. Funciones con retorno y sin retorno.

Código:

Documentación de mi código:

- **Amarillo** - Comentarios
- **Verde** - Librerías
- **Turquesa** - Funciones
- **Azul** - Condiciones
- **Rosa** - Bucles/Ciclos

```
// Practica 4 Funciones Con y Sin Retorno
// Teresa Rivas Gomez 372565
// Sep - 13 - 2023

// 1. Fibonacci sin Recursión: Crea un programa que calcule y muestre los primeros n
// términos de la serie de Fibonacci sin utilizar recursión.
// 2. Fibonacci con Recursión: Crea un programa en C que calcule y muestre el término
// n de la serie de Fibonacci utilizando una función recursiva.
// 3. Triángulo de Pascal: Crea un programa en C que genere y muestre las primeras n
// filas del Triángulo de Pascal.

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
// funciones
int msges();
void menu();
// funciones
int fibonacci_con(int i);
void fibonacci_sin();
void triangulo_de_pascal(void);
// funcion principal
int main()
{
    menu();
    return 0;
}
// menu que ve el usuario para elegir el problema a realizar
int msges()
{
    system ("CLS");
    int op;
    printf ("\n M E N U \n");
    printf ("1.- FIBONACCI CON RETORNO \n");
    printf ("2.- FIBONACCI SIN RETORNO \n");
    printf ("3.- TRIANGULO DE PASCAL \n");
    printf ("0.- SALIR \n");
    printf ("ESCOGE UNA OPCION: ");
    scanf ("%d",&op);
    return op;
}
```

```
}  
// para ejecutar diferentes opciones que hay  
void menu()  
{  
    int op;  
    do{  
        op=msges();  
        switch (op) // ejecucion de los diferentes casos  
        {  
            case 1:  
                system ("CLS");  
                int n, i;  
                printf("Ingrese cuantas veces desea que se ejecute Fibonacci: "); //  
                pedir a usuario un numero  
                scanf("%d",&n);  
  
                for( i = 0 ; i < n ; i++ ) // ciclo que manda a llamar a la formula del  
                fibonacci si se cumple  
                {  
                    printf("%d, ",fibonacci_con(i));  
                }  
                char respuesta;  
                printf("\n\nDeseas calcular otro valor? (S/N): "); // condicion para  
                saber que ejecutar dependiendo la respuesta del usuario  
                scanf(" %c", &respuesta);  
                if (respuesta == 'S' || respuesta == 's') // puse las dos S y s para  
                menor margen de error  
                {  
                    continue; // Volver al menu si desea calcular otro valor  
                }  
                else  
                {  
                    printf("Saliste del programa\n");  
                    exit(0);  
                }  
                system ("PAUSE");  
                break;  
            case 2:  
                fibonacci_sin();  
                break;  
            case 3:  
                triangulo_de_pascal();  
                break;  
        }  
    }while (op != 0); // este bucle se hace para seguir haciendose hasta que le den a  
    la opcion 0 de salir  
}  
// calcula y devuelve el termino de la i en el fibonacci utilizando la recursion  
int fibonacci_con(int i)  
{  
    if(i == 0)  
    {  
        return 0;  
    }  
    else if(i==1)  
    {
```

```
        return 1;
    }
    else
    {
        return (fibonacci_con(i-1)+fibonacci_con(i-2)); // operacion para el fibonacci
    }
}
// calcula y muestra la serie fibonacci usando un ciclo for
void fibonacci_sin()
{
    system ("CLS");
    int a, s, r, v, contador;
    printf("FIBONACCI CON FOR\n");
    r = 0;
    a = -1;
    s = 1;
    printf("Ingrese cuantas veces desea que se ejecute Fibonacci: ");
    scanf("%d", &v);
    for(contador=0;contador<v;contador=contador+1) // ciclo para realizar la suma del
    fibonacci
    {
        r = a + s;
        printf("%d, ",r);
        a = s;
        s = r;
    }
    // Preguntar si desea ejecutar de nuevo el programa
    char respuesta;
    printf("\n\nDeseas calcular otro valor? (S/N): ");
    scanf(" %c", &respuesta);
    if (respuesta == 'S' || respuesta == 's')
    {
        fibonacci_sin();
    }
    else
    {
        printf("Saliste del programa.\n");
    }
}
// calcula el triangulo de pascal dependiendo la cantidad de numeros que le pidas
void triangulo_de_pascal(void)
{
    system ("CLS");
    int num, resultado, i;
    resultado = 0;

    printf("TRIANGULO DE PASCAL\n");
    printf("Introduce la cantidad de numeros para el triangulo de pascal: ");
    scanf("%d", &num);
    for (i = 0; i <= num; i++) // para generar los numeros de la secuencia
    {
        resultado = pow(2, i); // calcular potencia a la 2
        printf("%d ", resultado);
    }
    // Preguntar si desea ejecutar de nuevo el programa
    char respuesta;
```

```

graph TD
    void_menu([void menu()]) --> int_op([int op])
    int_op --> op_menu[op=menu()]
    op_menu --> op_switch{op}
    op_switch -- 1 --> system_CLS_1[system "CLS"]
    op_switch -- 2 --> fibonacci_sin[void fibonacci_sin()]
    op_switch -- 3 --> triangulo_de_pascal[void triangulo_de_pascal()]
    op_switch --> int_n_i[int n, i]
    int_n_i --> print_ingresa[print "Ingrese cuantas veces desea que se ejecute Fibonacci"]
    print_ingresa --> scanf_n[scanf "%d", &n]
    scanf_n --> i_0[i = 0]
    i_0 --> i_n{ i < n }
    i_n -- True --> print_factorial[print "%d", fibonacci_con[i]]
    print_factorial --> i_plus[i++]
    i_plus --> i_n
    i_n -- False --> char_respuesta[char respuesta]
    char_respuesta --> print_otro_valor[print "\n\nDeseas calcular otro valor? (S/N): "]
    print_otro_valor --> scanf_respuesta[scanf "%c", &respuesta]
    scanf_respuesta --> respuesta_valida{respuesta == 'S' || respuesta == 's'}
    respuesta_valida -- True --> i_n
    respuesta_valida -- False --> print_salida[print "Saliste del programa\n"]
    print_salida --> exit_0[exit 0]
    exit_0 --> system_PAUSE[system "PAUSE"]
    system_PAUSE --> op_ne_0{op != 0}
    op_ne_0 -- True --> op_menu
    op_ne_0 -- False --> int_main([int main()])
    int_main --> menu_2([menu()])
    menu_2 --> i_0_2{i = 0}
    i_0_2 --> i_ne_1{i != 1}
    i_ne_1 -- True --> fibonacci_con_i_1[void fibonacci_con(i-1) fibonacci_con(i-2)]
    i_ne_1 -- False --> i_1([1])
    i_1 --> i_ne_1
    i_0_2 -- True --> i_0_3([0])
    i_0_3 --> i_ne_1
  
```



