



## Actividad | 3 | Código en Lenguaje C

### Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software

---



academias global

TUTOR: Sandra Luz Lara Devora.

---

ALUMNO: Aldair Quiñonez González.

---

FECHA: 19 de abril del 2025.

---

## Índice

Índice.....	1
1    Introducción .....	2
2    Descripción .....	2
3    Justificación .....	3
4    Desarrollo.....	4
4.1    Punto 1. Números Primos. ....	4
4.2    Punto 2. Números pares e impares.....	5
4.3    Punto 3. Números al revés o invertidos. ....	5
5    Conclusión .....	6
6    Referencias.....	7

## **1 Introducción**

En esta actividad desarrollé tres calculadoras utilizando el lenguaje de programación C, con el objetivo de aplicar y reforzar mis conocimientos en lógica de programación, estructuras de control y manipulación de datos. Cada una de las calculadoras cumple una función específica. La primera está diseñada para identificar si un número ingresado por el usuario es primo o no, utilizando una estructura condicional y un bucle para comprobar los divisores posibles. La segunda calculadora determina si un número es par o impar, empleando una operación matemática simple con el operador módulo. Por último, la tercera calculadora invierte un número de cuatro dígitos, descomponiéndolo y reordenando sus cifras utilizando operaciones aritméticas. Esta actividad me permitió profundizar en el uso de variables, ciclos, condicionales y la lógica necesaria para resolver problemas matemáticos básicos mediante programación. Además, fortalecí mi capacidad para estructurar el código de manera ordenada y eficiente, y comprendí mejor cómo interactuar con el usuario mediante entradas y salidas en C.

## **2 Descripción**

La actividad consistió en el desarrollo de tres programas utilizando el lenguaje de programación C, enfocados en resolver problemas matemáticos básicos mediante el uso de estructuras de control, operadores y lógica algorítmica. La primera calculadora fue diseñada para determinar si un número ingresado por el usuario es primo. Para ello, se implementó un ciclo que evalúa si el número tiene más de dos divisores, y en base a esa condición se da el resultado. La segunda calculadora identifica si un número es par o impar, utilizando el operador módulo (%) para analizar el residuo de la división entre 2. Esta operación permite saber rápidamente si el

número cumple con la condición de ser divisible exactamente por dos. Finalmente, la tercera calculadora invierte un número de cuatro cifras. Esto se logra extrayendo cada dígito mediante divisiones y módulos, para luego reorganizarlos en orden inverso. A través de esta actividad, se fortalecieron habilidades fundamentales en programación estructurada, como el manejo de bucles, condicionales, entrada y salida de datos, así como el uso correcto de variables y operadores. También se trabajó la lógica necesaria para transformar ideas en soluciones computacionales funcionales. Esta práctica fue útil para aplicar conceptos teóricos y mejorar la comprensión del lenguaje C en contextos prácticos.

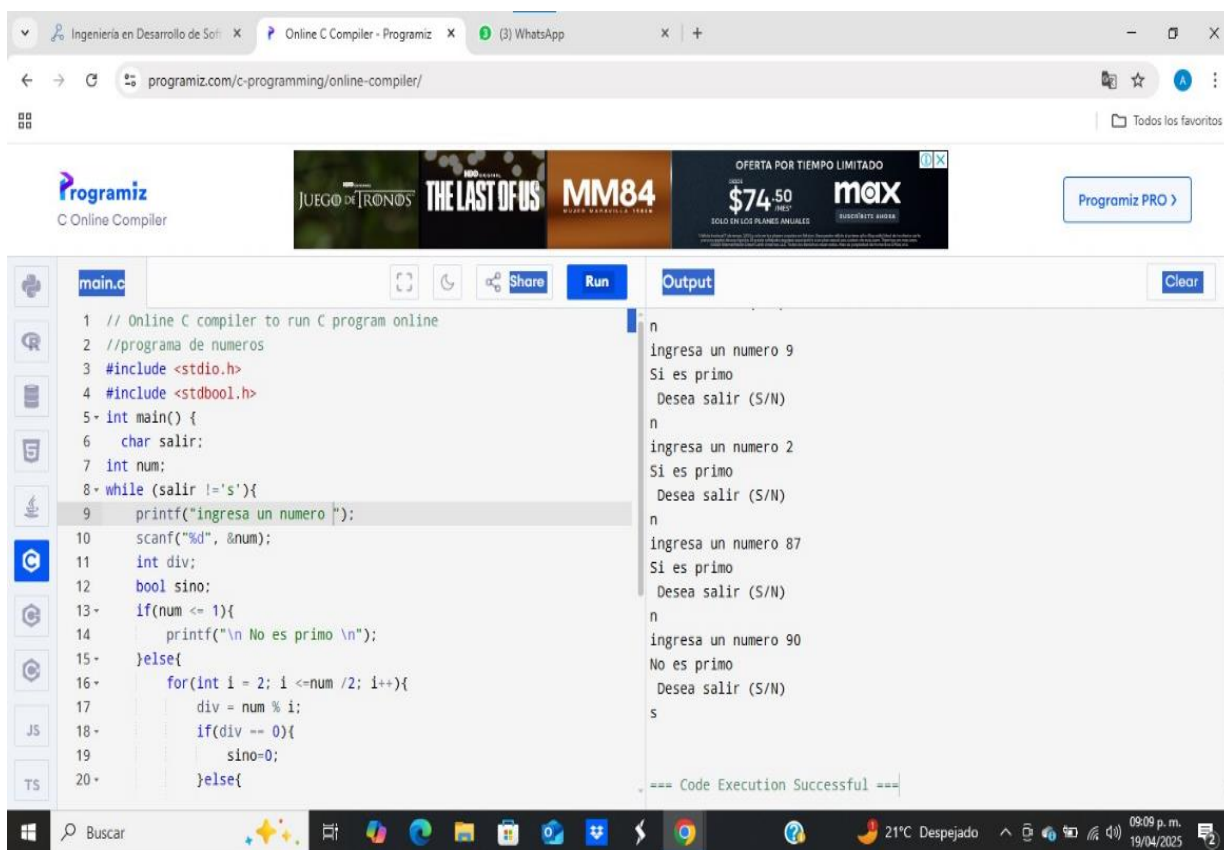
### **3 Justificación**

La realización de esta actividad se justifica como una forma práctica de aplicar los conceptos fundamentales del lenguaje de programación C, enfocados en el uso de estructuras de control como if, else y do while. Estas estructuras permiten al programa tomar decisiones y repetir acciones según las condiciones establecidas, lo cual es esencial en el desarrollo de cualquier aplicación lógica. En la calculadora de números primos, por ejemplo, se utilizó la estructura if-else para determinar si un número cumple con la condición de ser primo. Además, se implementó un bucle do while para permitir al usuario repetir el proceso tantas veces como desee. También se trabajó con la notación factorial ( $n!$ ), especialmente para evaluar de manera más eficiente algunos cálculos relacionados con los números primos, fortaleciendo la comprensión de operaciones matemáticas avanzadas dentro del contexto de programación. Esta actividad no solo mejora la lógica y el pensamiento computacional, sino que también permite consolidar el conocimiento de la sintaxis y estructura del lenguaje C.

## 4 Desarrollo

Adjunto evidencia del desarrollo que realice en lenguaje C, para la realización de tres calculadoras donde una es en cuanto a números primos, otra a números pares e impares y otra donde da el resultado de cuatro dígitos de forma invertida.

### 4.1 Punto 1. Números Primos.



The screenshot displays the Programiz Online C Compiler interface. The browser address bar shows the URL `programiz.com/c-programming/online-compiler/`. The page header includes the Programiz logo and a banner for MM84. The main content area is divided into two panels: a code editor on the left and an output window on the right.

The code editor shows a C program named `main.c` with the following content:

```
1 // Online C compiler to run C program online
2 //programa de numeros
3 #include <stdio.h>
4 #include <stdbool.h>
5 int main() {
6     char salir;
7     int num;
8     while (salir != 's'){
9         printf("ingresa un numero ");
10        scanf("%d", &num);
11        int div;
12        bool sino;
13        if(num <= 1){
14            printf("\n No es primo \n");
15        }else{
16            for(int i = 2; i <=num /2; i++){
17                div = num % i;
18                if(div == 0){
19                    sino=0;
20                }else{
```

The output window displays the program's execution results:

```
n
ingresa un numero 9
Si es primo
Desea salir (S/N)
n
ingresa un numero 2
Si es primo
Desea salir (S/N)
n
ingresa un numero 87
Si es primo
Desea salir (S/N)
n
ingresa un numero 90
No es primo
Desea salir (S/N)
s
=== Code Execution Successful ===
```

The Windows taskbar at the bottom shows the system clock as 09:09 p.m. on 19/04/2025, with a temperature of 21°C and the location Despejado.

## 4.2 Punto 2. Números pares e impares.

```

1 // Online C compiler to run C program online
2 //programa de numeros
3 #include <stdio.h>
4
5 int main() {
6     int valor = 0;
7     int n;
8     do{
9         printf("Ingrese un numero \n");
10        scanf("%d", &n);
11        if(n % 2 == 0){
12            printf("%d es par \n", n);
13        }else{
14            printf("%d es impar \n", n);
15        }
16        valor = valor + 1;
17    }while(valor < 10);
18    printf("*****termino el ciclo*****");
19    return 0;
20 }
  
```

Output

```

67
67 es impar
, n:ingrese un numero
67
67 es impar
, n:ingrese un numero
56
56 es par
, n:ingrese un numero
3
3 es impar
, n:ingrese un numero
3
3 es impar
, n*****termino el ciclo*****
=== Code Execution Successful ===
  
```

## 4.3 Punto 3. Números al revés o invertidos.

```

1 // Online C compiler to run C program online
2 //programa de numeros
3 #include <stdio.h>
4
5 int main() {
6     int n, resto, invertido = 0;
7     printf("Ingrese un numero ");
8     scanf("%d", &n);
9     while(n != 0){
10        resto = n % 10;
11        n = n / 10;
12        invertido = invertido * 10 + resto;
13    }
14    printf("El numero invertido es ");
15    printf("%d \n", invertido);
16    return 0;
17 }
  
```

Output

```

Ingrese un numero 6789
El numero invertido es 9876

=== Code Execution Successful ===
  
```

## 5 Conclusión

En conclusión, esta actividad representó una valiosa oportunidad para aplicar de manera práctica los conocimientos adquiridos sobre programación en lenguaje C. A través del desarrollo de tres calculadoras una para identificar números primos, otra para determinar si un número es par o impar, y una tercera para invertir un número de cuatro cifras logré fortalecer mi comprensión sobre estructuras de control como if, else y do while, así como el manejo de operadores matemáticos y lógicos. Además, la incorporación de la notación factorial ( $n!$ ) en algunos procesos permitió explorar con mayor profundidad la lógica detrás de ciertos cálculos, especialmente en la identificación de números primos. Esta experiencia también me ayudó a mejorar mi habilidad para organizar y estructurar el código de forma clara y eficiente, algo fundamental en el desarrollo de programas funcionales y fáciles de entender. La práctica constante mediante este tipo de ejercicios facilita la identificación de errores, el diseño de soluciones algorítmicas y el fortalecimiento del pensamiento computacional. En general, puedo concluir que este trabajo no solo consolidó mis conocimientos teóricos, sino que también me permitió experimentar con el lenguaje C en un entorno práctico, incrementando mi confianza y habilidad para resolver problemas mediante programación estructurada.

## 6 Referencias

<https://www.programiz.com/c-programming/online-compiler/>

<https://openwebinars.net/blog/que-es-c/>