

# **Actividad 3 - Código en Lenguaje C**

## **Introducción al Desarrollo de**

## **Software**

### **Ingeniería en Desarrollo de Software**

**Tutor: Felipe de Jesús Araux López.**

**Alumno: Uziel Abisai Martinez Oseguera.**

**Fecha: 01/04/2023.**

## Índice

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>DESCRIPCIÓN .....</b>	<b>2</b>
<b>JUSTIFICACIÓN .....</b>	<b>3</b>
<b>INVESTIGACIÓN.....</b>	<b>4</b>
¿QUÉ ES EL LENGUAJE C? .....	4
APLICACIÓN DEL LENGUAJE C EN EL MUNDO LABORAL .....	5
<b>DESARROLLO .....</b>	<b>6</b>
PROGRAMA NÚMEROS PRIMOS .....	6
PROGRAMA NÚMEROS PARES E IMPARES.....	9
PROGRAMA NÚMEROS INVERTIDOS .....	11
<b>CONCLUSIÓN.....</b>	<b>14</b>
<b>REFERENCIAS .....</b>	<b>15</b>

## **Introducción**

La actividad consistió en el desarrollo de tres programas matemáticos utilizando el lenguaje de programación C, con el objetivo de implementarlos en colegios y escuelas públicas. Para ello, se debió realizar un algoritmo, un diagrama de flujo y finalmente la codificación en C de cada programa. Los programas desarrollados fueron: Primos, Par/Impar y Al Revés, los cuales permiten al usuario determinar si un número es primo o no, identificar si un número es par o impar y también invertir un número entero de 4 dígitos. Estos programas fueron diseñados para ser utilizados de manera clara y concisa, con el fin de facilitar su comprensión y uso por parte de los estudiantes.

## Descripción

En la actividad se desarrollaron tres programas matemáticos utilizando el lenguaje de programación C, con el objetivo de implementarlos en colegios y escuelas públicas. Para cada programa, se comenzó por realizar un algoritmo que permitiera comprender la lógica del programa y cómo se manipulaban los datos ingresados por el usuario. Luego, se creó un diagrama de flujo que representaba de manera gráfica el algoritmo previamente diseñado.

Una vez que se tuvo el algoritmo y el diagrama de flujo, se procedió a la codificación del programa en C. Para ello, se utilizaron estructuras de control, como bucles y condicionales, y operaciones matemáticas para realizar las operaciones necesarias en cada programa. Se aseguró de que los programas fueran sencillos y fáciles de usar, para que los estudiantes puedan utilizarlos sin dificultad.

Cada uno de los programas matemáticos tiene una funcionalidad específica. El programa Primos permite determinar si un número es primo o no, el programa Par/Impar permite identificar si un número es par o impar y el programa Al Revés permite invertir un número entero de 4 dígitos. Estos programas fueron diseñados pensando en la facilidad de uso para los estudiantes, con la finalidad de ayudarlos a comprender mejor conceptos matemáticos y fomentar su interés por la programación.

### **Justificación**

Los programas matemáticos creados en esta actividad tienen una gran importancia en el contexto educativo actual, ya que promueven el aprendizaje de conceptos matemáticos de manera lúdica y atractiva para los estudiantes. Además, su implementación en colegios y escuelas públicas puede mejorar la calidad de la enseñanza y la comprensión de los estudiantes en el área de matemáticas.

El programa Primos, por ejemplo, es una herramienta útil para que los estudiantes aprendan a identificar números primos, lo que puede ayudarles a desarrollar habilidades lógicas y matemáticas. El programa Par/Impar permite a los estudiantes comprender mejor la paridad de los números y, por lo tanto, facilita su comprensión de otros conceptos matemáticos más complejos. Por último, el programa Al Revés, les permite a los estudiantes practicar el uso de la lógica y la aritmética al invertir los números enteros de 4 dígitos.

En resumen, estos programas matemáticos pueden ser una herramienta valiosa para el aprendizaje de los estudiantes y, por lo tanto, su creación y aplicación en colegios y escuelas públicas es justificable y recomendable.

## **Investigación**

### **¿Qué es el lenguaje C?**

El lenguaje C es un lenguaje de programación de alto nivel y propósito general que fue desarrollado originalmente en la década de 1970 por Dennis Ritchie en los laboratorios Bell de AT&T. Es un lenguaje de programación muy popular debido a su eficiencia, portabilidad y capacidad para manipular directamente los recursos del hardware de una computadora.

C es un lenguaje estructurado, lo que significa que utiliza una sintaxis clara y precisa que permite a los programadores escribir código de manera fácil de entender y mantener. Además, C cuenta con una gran cantidad de bibliotecas y herramientas, lo que lo hace muy versátil y adecuado para una amplia gama de aplicaciones.

C es ampliamente utilizado en la industria del software para desarrollar sistemas operativos, compiladores, lenguajes de programación, aplicaciones de redes, sistemas embebidos y muchos otros tipos de software. También se utiliza en la enseñanza de la programación y es uno de los primeros lenguajes de programación que se enseñan en cursos universitarios de informática.

## **Aplicación del lenguaje C en el mundo laboral**

El lenguaje C tiene una amplia aplicación en el mundo laboral, ya que es utilizado en diversos campos de la industria del software, incluyendo:

1.- Desarrollo de sistemas operativos: El sistema operativo Unix y sus derivados (como Linux y macOS) están escritos en C, lo que muestra la capacidad del lenguaje para manejar tareas de bajo nivel y alta complejidad.

2.- Desarrollo de compiladores: Muchos compiladores de lenguaje de programación están escritos en C, incluyendo el compilador GCC (GNU Compiler Collection), que es ampliamente utilizado para compilar programas en lenguaje C y C++.

3.- Desarrollo de aplicaciones de sistemas embebidos: C es utilizado para desarrollar aplicaciones en sistemas embebidos, como controladores de dispositivos, sistemas de control de procesos, sistemas de automatización industrial y otros sistemas electrónicos.

4.- Desarrollo de aplicaciones de redes: C es utilizado en el desarrollo de aplicaciones de redes, como servidores web, servidores de correo electrónico y otros tipos de aplicaciones de red.

5.- Desarrollo de videojuegos: C es utilizado en el desarrollo de videojuegos, debido a su capacidad para manipular directamente los recursos de hardware, lo que permite a los programadores crear juegos con gráficos de alta calidad y una gran interactividad.

## Desarrollo

### Programa Números Primos

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

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

```
int main() {
```

```
    int num, i, raiz;
```

```
    printf("Ingrese un número entero para verificar si es primo: ");
```

```
    scanf("%d", &num);
```

```
    if (num <= 1) {
```

```
        printf("El número %d no es primo.\n", num);
```

```
        return 0;
```

```
    }
```

```
    if (num == 2 || num == 3) {
```

```
        printf("El número %d es primo.\n", num);
```

```
        return 0;
```

```
    }
```

```
    if (num % 2 == 0) {
```

```
        printf("El número %d no es primo.\n", num);
```

```
        return 0;
```



```
}
```

```
raiz = sqrt(num);
```

```
for (i = 3; i <= raiz; i += 2) {
```

```
    if (num % i == 0) {
```

```
        printf("El número %d no es primo.\n", num);
```

```
        return 0;
```

```
    }
```

```
}
```

```
printf("El número %d es primo.\n", num);
```

```
return 0;
```

```
}
```

The screenshot shows a C++ IDE with a file named `main.c`. The code implements a prime number checker. It includes `<stdio.h>` and `<math.h>`. The `main` function prompts the user to enter an integer. It handles edge cases for numbers less than or equal to 1, and numbers 2 and 3. For other numbers, it checks divisibility by 2 and then by odd numbers from 3 up to the square root of the input number. The output window shows the program's execution with the input 83, resulting in the message "El número 83 es primo."

```
main.c
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     int num, i, raiz;
6     printf("Ingrese un número entero para verificar si es primo: ");
7     scanf("%d", &num);
8
9     if (num <= 1) {
10        printf("El número %d no es primo.\n", num);
11        return 0;
12    }
13
14    if (num == 2 || num == 3) {
15        printf("El número %d es primo.\n", num);
16        return 0;
17    }
18
19    if (num % 2 == 0) {
20        printf("El número %d no es primo.\n", num);
21        return 0;
22    }
23
24    raiz = sqrt(num);
25
26    for (i = 3; i <= raiz; i += 2) {
27        if (num % i == 0) {
28            printf("El número %d no es primo.\n", num);
29            return 0;
30        }
31    }
32
33    printf("El número %d es primo.\n", num);
34    return 0;
35 }
```

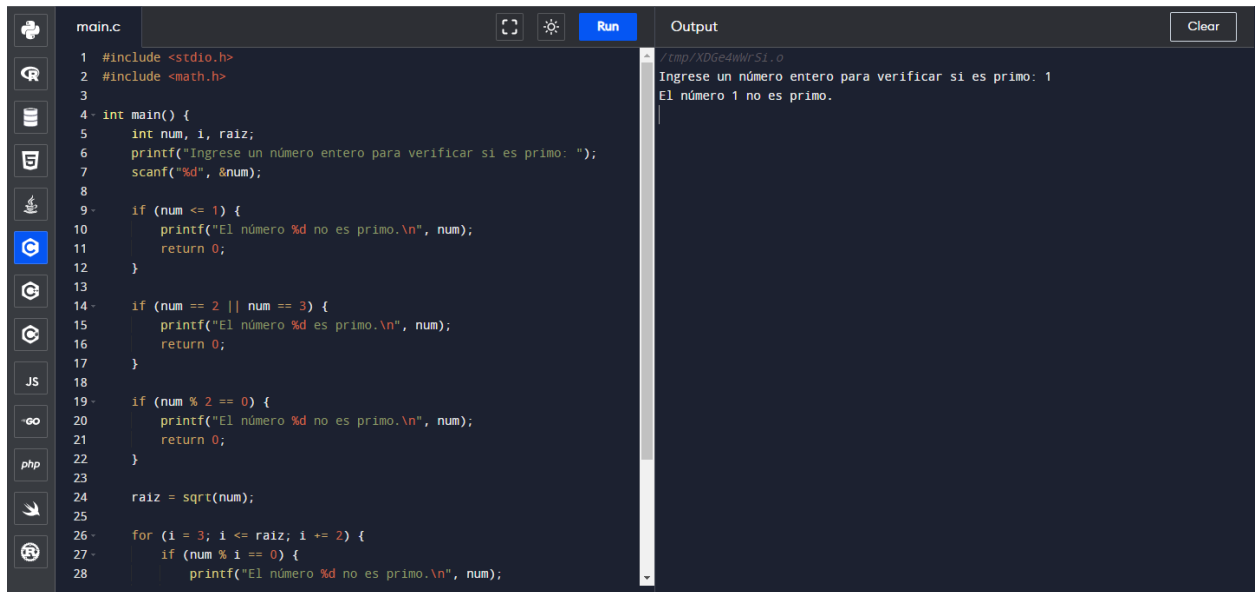
Output

```

/rmp/XDGedWlrS1.o
Ingrese un número entero para verificar si es primo: 83
El número 83 es primo.

```

Esperando a securepubads.g.doubleclick.net...



```

1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     int num, i, raiz;
6     printf("Ingrese un número entero para verificar si es primo: ");
7     scanf("%d", &num);
8
9     if (num <= 1) {
10        printf("El número %d no es primo.\n", num);
11        return 0;
12    }
13
14    if (num == 2 || num == 3) {
15        printf("El número %d es primo.\n", num);
16        return 0;
17    }
18
19    if (num % 2 == 0) {
20        printf("El número %d no es primo.\n", num);
21        return 0;
22    }
23
24    raiz = sqrt(num);
25
26    for (i = 3; i <= raiz; i += 2) {
27        if (num % i == 0) {
28            printf("El número %d no es primo.\n", num);

```

Output

```

/tmp/XDGe4wlrS1.o
Ingrese un número entero para verificar si es primo: 1
El número 1 no es primo.

```

El código presentado es un programa en lenguaje C que verifica si un número entero ingresado por el usuario es primo o no, utilizando el algoritmo de números primos.

En cuanto a la función que se está ejecutando, se trata de la función principal "main()" del programa. Esta función se encarga de leer el número ingresado por el usuario, aplicar el algoritmo de números primos para verificar si el número es primo o no, y finalmente imprimir un mensaje indicando si el número es primo o no.

Para realizar esta tarea, la función utiliza la biblioteca estándar "stdio.h" para realizar la entrada y salida de datos, y la biblioteca "math.h" para calcular la raíz cuadrada del número a verificar. Además, utiliza variables enteras para almacenar el número ingresado por el usuario, la raíz cuadrada del número, y un contador utilizado en el ciclo de comprobación de números primos.

## Programa Números Pares e Impares

```
#include <stdio.h>

int main() {

    int num, i;

    printf("Ingrese 10 números enteros para verificar si son pares o impares:\n");

    for (i = 1; i <= 10; i++) {

        printf("Número %d: ", i);

        scanf("%d", &num);

        if (num % 2 == 0) {

            printf("El número %d es par.\n", num);

        } else {

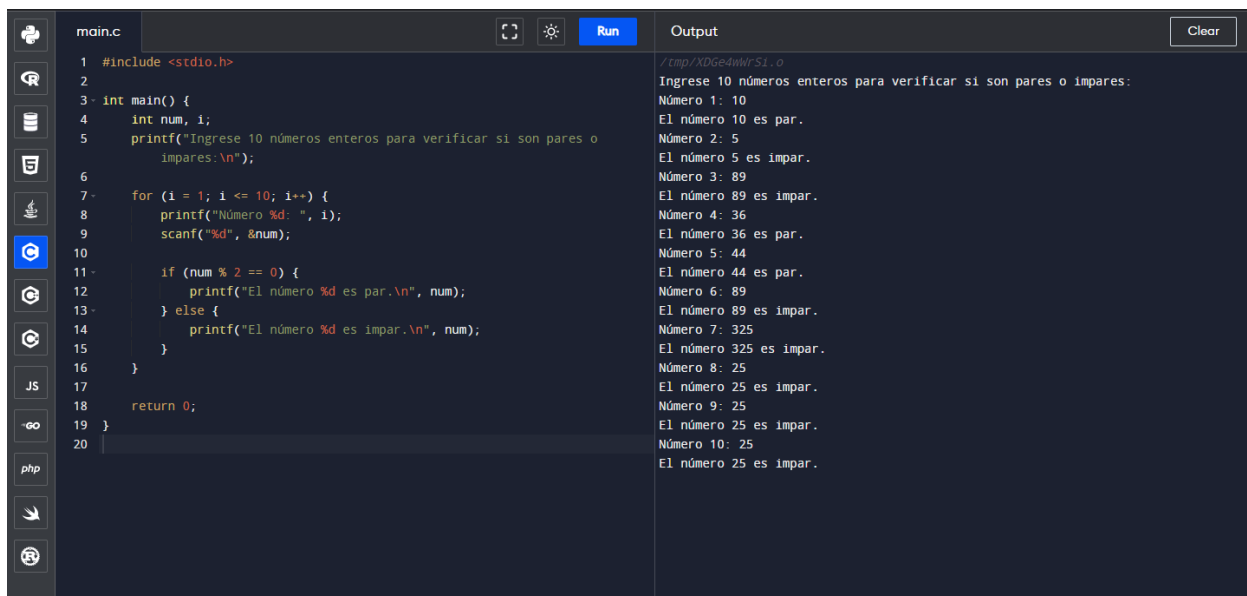
            printf("El número %d es impar.\n", num);

        }

    }

    return 0;

}
```



```

main.c
1 #include <stdio.h>
2
3 int main() {
4     int num, i;
5     printf("Ingrese 10 números enteros para verificar si son pares o
    impares:\n");
6
7     for (i = 1; i <= 10; i++) {
8         printf("Número %d: ", i);
9         scanf("%d", &num);
10
11         if (num % 2 == 0) {
12             printf("El número %d es par.\n", num);
13         } else {
14             printf("El número %d es impar.\n", num);
15         }
16     }
17
18     return 0;
19 }
20
Output
/tmp/XDg64WlrS1.o
Ingrese 10 números enteros para verificar si son pares o impares:
Número 1: 10
El número 10 es par.
Número 2: 5
El número 5 es impar.
Número 3: 89
El número 89 es impar.
Número 4: 36
El número 36 es par.
Número 5: 44
El número 44 es par.
Número 6: 89
El número 89 es impar.
Número 7: 325
El número 325 es impar.
Número 8: 25
El número 25 es impar.
Número 9: 25
El número 25 es impar.
Número 10: 25
El número 25 es impar.

```

En este programa, se utiliza un bucle "for" para obtener 10 números enteros ingresados por el usuario, y se utiliza la operación módulo "%" para determinar si cada número es par o impar. Si el resultado de la operación es igual a cero, el número se considera par y se imprime un mensaje indicando esto. De lo contrario, el número se considera impar y se imprime un mensaje indicando esto.

La biblioteca estándar "stdio.h" se utiliza para realizar la entrada y salida de datos, mientras que las variables enteras se utilizan para almacenar el número ingresado por el usuario y un contador utilizado en el bucle "for".

**Programa Números Invertidos**

```
#include <stdio.h>

int main() {

    int num, dig1, dig2, dig3, dig4, inv_num;

    while (1) {

        printf("Ingrese un número entero de 4 dígitos: ");

        scanf("%d", &num);

        if (num < 1000 || num > 9999) {

            printf("El número ingresado no es de 4 dígitos. Intente de nuevo.\n");

            continue;

        }

        dig1 = num % 10;

        num /= 10;

        dig2 = num % 10;

        num /= 10;

        dig3 = num % 10;

        num /= 10;

        dig4 = num % 10;
```

```
inv_num = dig1 * 1000 + dig2 * 100 + dig3 * 10 + dig4;
```

```
printf("El número invertido es: %d\n", inv_num);
```

```
}
```

```
return 0;
```

```
}
```

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int num, dig1, dig2, dig3, dig4, inv_num;
5
6     while (1) {
7         printf("Ingrese un número entero de 4 dígitos: ");
8         scanf("%d", &num);
9
10        if (num < 1000 || num > 9999) {
11            printf("El número ingresado no es de 4 dígitos. Intente de nuevo.\n");
12            continue;
13        }
14
15        dig1 = num % 10;
16        num /= 10;
17        dig2 = num % 10;
18        num /= 10;
19        dig3 = num % 10;
20        num /= 10;
21        dig4 = num % 10;
22
23        inv_num = dig1 * 1000 + dig2 * 100 + dig3 * 10 + dig4;
24
25        printf("El número invertido es: %d\n", inv_num);
26    }
27}
```

Output

```

/tmp/XDGedWwRS1.o
Ingrese un número entero de 4 dígitos: 12
El número ingresado no es de 4 dígitos. Intente de nuevo.
Ingrese un número entero de 4 dígitos: 1234
El número invertido es: 4321
Ingrese un número entero de 4 dígitos: |

```

Esperando a securepubads.googleclicknet...

El programa utiliza un bucle while para permitir que el usuario ingrese múltiples números. En cada iteración, se solicita al usuario que ingrese un número de 4 dígitos y se verifica que sea un número entero de 4 dígitos. Si no lo es, se muestra un mensaje de error y se solicita que se ingrese otro número.

Luego, el número se divide en sus cuatro dígitos individuales utilizando la operación módulo y la división entera. A continuación, los dígitos se reordenan en orden inverso y se combinan para formar el número invertido.

Finalmente, se muestra el número invertido al usuario. El programa vuelve al paso 1 para permitir que el usuario ingrese otro número.

## Conclusión

En conclusión, es esencial comprender las operaciones matemáticas detrás de un problema antes de comenzar a codificar, ya que esto permite tener una mejor comprensión del problema y encontrar la mejor solución posible. Los algoritmos y diagramas de flujo son herramientas útiles para ayudar a visualizar el proceso detrás de un problema y encontrar posibles errores antes de comenzar a escribir código.

Durante estas tres actividades, aprendimos a desarrollar programas en lenguaje C para resolver problemas matemáticos específicos, como identificar números primos, determinar si un número es par o impar y revertir un número de cuatro dígitos. Aprendimos a construir algoritmos y diagramas de flujo para estos programas y a traducirlos a código funcional en C. También aprendimos sobre la importancia de la eficiencia y la optimización en el desarrollo de software.

En general, estas actividades nos proporcionaron una base sólida para continuar aprendiendo sobre programación y cómo aplicarla en diferentes campos de trabajo. A medida que continuamos desarrollando habilidades en lenguajes de programación, podemos seguir mejorando nuestras capacidades para resolver problemas y construir soluciones innovadoras en el mundo laboral.



## **Referencias**

No hay referencias.