

## **Actividad | # 3| Nombre de la**

### **actividad: Etapa 3 – Código en Lenguaje C**

#### **Nombre del curso**

**Introducción al Desarrollo de Software**

**Ingeniería en Desarrollo de Software**



TUTOR: Dra. Sandra Luz Lara Devora

ALUMNO: Jesús Terán Díaz

FECHA:17/04/2025

## Índice

Introducción.....	3
Descripción.....	4
Justificación.....	5
Desarrollo.....	6
Numero Primo.....	6
Numero par e impar.....	7
Numero invertido.....	9
Conclusión.....	11
Referencias.....	12

## **Introducción**

En la presente actividad se ha necesitado la programación de tres distintas calculadoras las cuales serán implementadas en colegios públicos y privados, el cual su objetivo principal es solucionar distintas problemáticas que se están presentando.

Ya planificado y ordenado esto se realizará la programación en cada situación con la finalidad de saber las características, funciones a realizar, objetivos y posibles escenarios que se puedan presentar en la elaboración de estos, llevándose acabo con el uso de variables y analizando el objetivo que se desea cumplir.

Por lo tanto, ya conociendo el contexto de la situación deberemos elaborar algoritmos, diagramas de flujo y la codificación en lenguaje c para dar solución a la problemática presentada anteriormente.

En esta actividad se ve la importancia de cada proceso y además en que consiste cada uno de estos para adentrarnos mas a fondo y adquirir nuevas opciones para dar solución a este tipo de problemáticas.

Estudiar y aprender a programar en c permite desarrollar una comprensión solida de los principios básicos de la programación estructurada, manejo de memoria, estructura de control y algoritmos. Además, facilita el desarrollo de una lógica de programación rigurosa, necesaria para resolver los problemas computacionales de manera efectiva.

En este contexto, se presentan 3 programas básico como parte de la introducción practica al lenguaje, una calculadora aritmética simple un verificador de numero primo y par e impar, y un inversor de número. Estos ejercicios tienen como finalidad aplicar los conceptos fundamentales de entrada y salida, estructura condicionales y ciclos, promoviendo una base solida para el aprendizaje de algoritmos más complejos.

## Descripción

El objetivo principal de la presente actividad es programar 3 tipos de calculadoras las cuales se considerarán implementar tanto en colegios privados como escuelas públicas en las cuales cada una tiene sus distintas enseñanzas y exigencias para su uso.

En la primera calculadora llevará por nombre Primos, y el objetivo será al ingresar un numero en el cual deberá imprimir el mensaje si es primo, en el caso contrario el número ingresado no es primo.

La segunda calculadora lleva por nombre Par e Impar, el cual es el objetivo es ingresar 10 números ya sean pares o impares, por ejemplo, si ingresamos el numero 7 el programa deberá indicar que es un numero impar, y si hablamos del número 4 el programa deberá indicar que se trata de un numero par. De 10 números enteros se determinará cuáles son pares y cual son impares.

La última calculadora que corresponde a la tercera se denomina número invertido, aquí el objetivo es que el usuario ingrese un numero de 4 dígitos y que sean números enteros, ya que este programa se encargara de invertir los números al revés. Por ejemplo, si ingresamos el número 8642, el programa matemático deberá invertirlo al 2468.

La finalidad de este proyecto es aplicar los fundamentos de lenguaje de programación C mediante la creación de 3 programas básico, los cuales permiten afianzar el uso de estructura de control, entrada y salida de datos y lógica algorítmicas.

Por lo tanto cada uno de estos ejercicios permite al estudiante comprender la lógica detrás de problemas cotidianos y su resolución computacional fomentando así habilidades esenciales respecto a la programación estructurada.

## **Justificación**

En el presente apartado de la actividad se dará a entender y compartir la importancia de estos procesos. En el desarrollo de habilidades respecto a la programación estructurada es fundamental para la formación de cualquier profesional en el área de ingeniería, tecnología o ciencia computacionales. En el lenguaje C, por su cercanía al hardware y su estructura lógica clara, representa una excelente herramienta para introducir al estudiante al pensamiento algorítmico, así como la resolución de problemas computacionales.

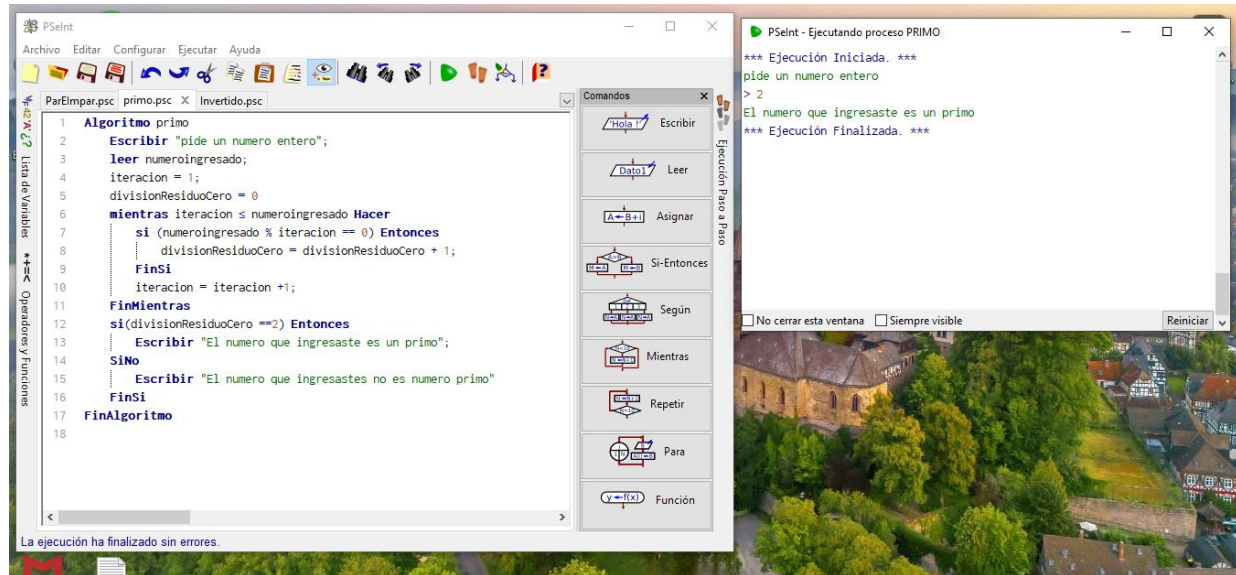
En la creación de programas como una calculadora básica, un verificador de número primo, par e impar y un inversor de número no solo permite aplicar los conceptos teóricos aprendidos en clase, sino que también estimula el análisis, la lógica matemática y la capacidad de traducir un problema real a un lenguaje computacional.

Este trabajo contribuye al desarrollo de competencias esenciales en la programación básica, sirviendo como base para enfrentar problemas más complejos que se ven en la vida cotidiana en un futuro profesional y académico.

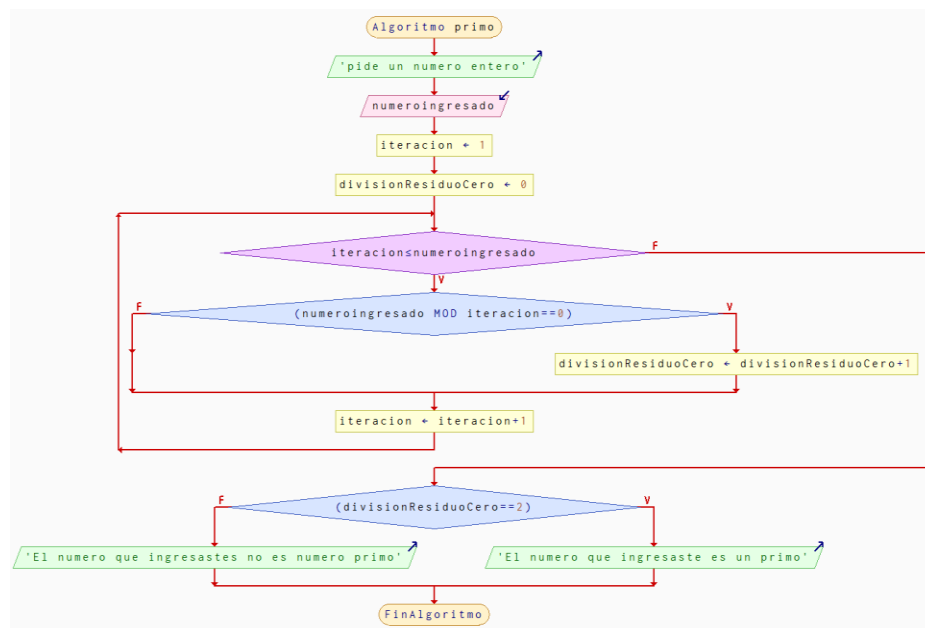
## Desarrollo

### Números primos.

#### Algoritmo



#### Diagrama de flujo



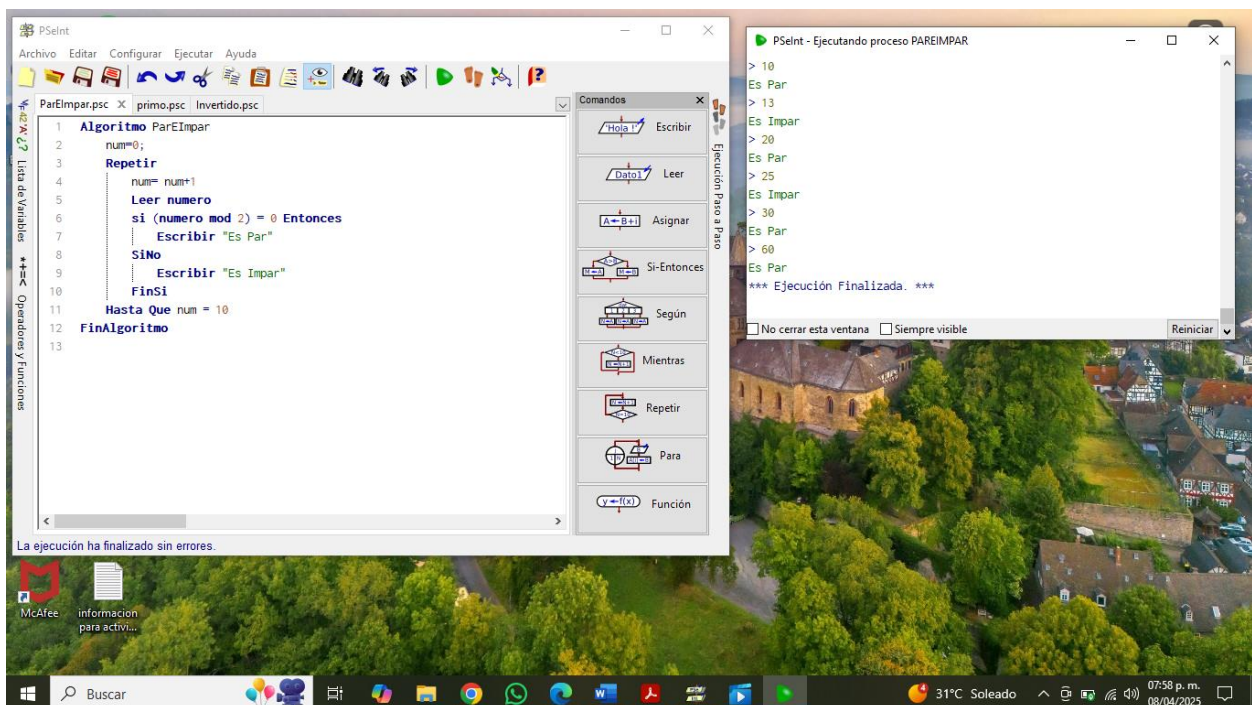
# Codificación

```
1 // Online C compiler to run C program online
2 //programa de numero primo
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){
```

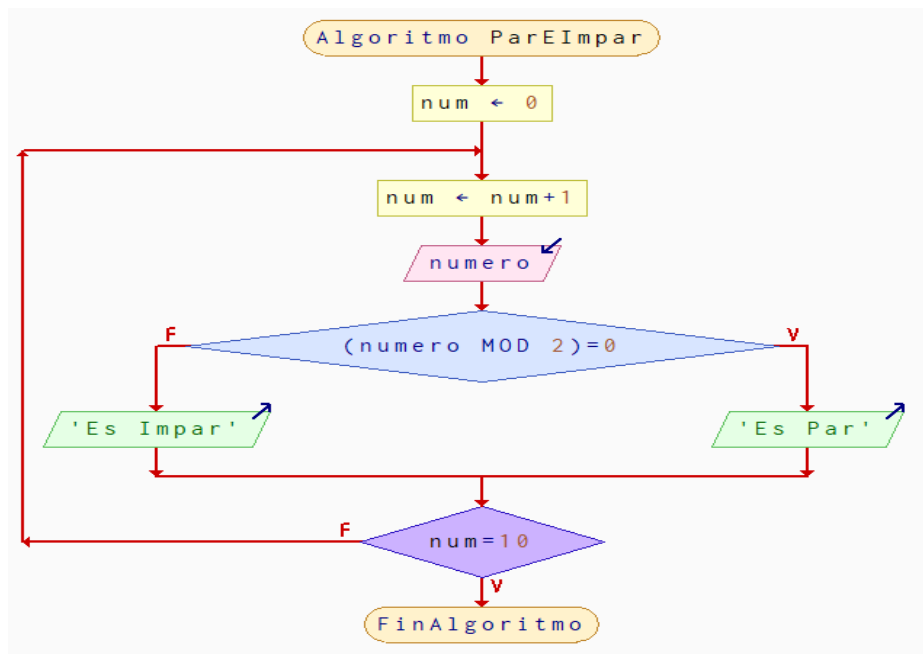
Ingresa un numero5  
si es primo  
Desea salir (s/n)  
s  
  
=== Code Execution Successful ===

## Número par e impar.

## Algoritmo



## Diagrama de flujo



## Codificación

```
main.c
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");
15-        }
16-        valor = valor + 1;
17-    }while(valor < 10);
18-    printf("*****termino el ciclo*****");
19-    return 0;
20- }
21- }
```

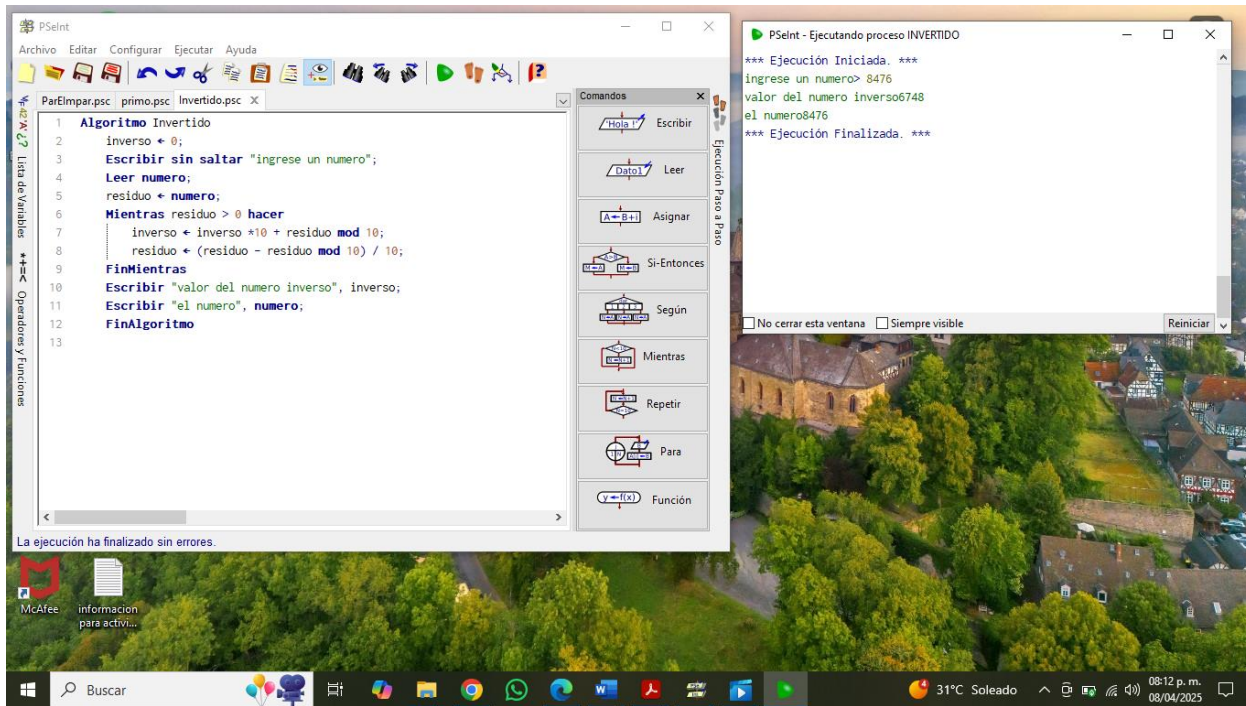
Output

```
Ingrese un numero
4
4 es par
Ingrese un numero
5
5 es impar
Ingrese un numero
```

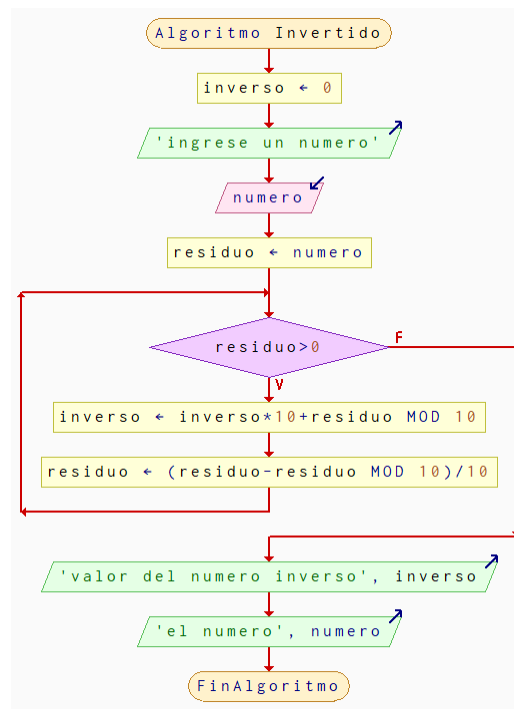


## Algoritmo números invertidos.

### Algoritmo



### Diagrama de flujo



# Codificación

The screenshot displays the Programiz Online C Compiler web application. The browser's address bar shows the URL `programiz.com/c-programming/online-compiler/`. The page header includes the Programiz logo, a Mercado Play advertisement, and a 'Programiz PRO' button. The main interface is divided into two sections: a code editor on the left and an output window on the right.

**Code Editor (main.c):**

```
1 // Online C compiler to run C program online
2 // programa Invertido
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 numeor invertido es ");
15     printf("%d \n", invertido);
16     return 0;
17 }
18
```

**Output Window:**

```
ingrese un numero 1234
El numeor invertido es 4321

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

The Windows taskbar at the bottom shows the system clock as 08:15 p.m. on 17/04/2025.

## **Conclusión**

En resumen, esta actividad nos adentramos en los distintos métodos para la solución de un problema como lo es un algoritmo, un diagrama de flujo y la codificación en lenguaje C, llegando al punto de facilitar la solución la cual en ocasiones es un problema para poder llegar a la solución.

Observando el proceso de cada herramienta en donde nos adentramos a detalle para poder conocer el funcionamiento y como poder planear un método para obtener el resultado esperado y poder debatir con estos problemas que comúnmente se pueden presentar.

El algoritmo, diagrama de flujo y la codificación es una guía para la creación de aplicaciones ya que es la base en lo que va dirigido, así como en donde se pueda identificar fácilmente el problema a resolver y los procesos que se tiene que seguir para poder obtener un resultado efectivo favorable.

Este proyecto permitió no solo aplicar conceptos teóricos de programación, sino también llevar a cabo el desarrollo de soluciones practicas que pueden ser reutilizadas o ampliadas en aplicaciones más complejas.

## Referencia

***PSEInt.* (s. f.). <https://pseint.sourceforge.net/>**

Link github  
<https://github.com/yishus-ops/Actividad-2.git>