

Actividad 3: Códigos en Lenguaje C

Materia: Introducción al Desarrollo de
Software

Ingeniería en Desarrollo de Software



TUTOR: **Sandra Lara Devora**

ALUMNO: **Luis David Ruiz Villanueva**

FECHA: **09 de agosto 2025**

Índice

Contenido

Introducción:	1
Descripción:	3
Justificación:	5
Desarrollo:	7
Algoritmos:	7
Numero primos:	7
Números parEimpar:	9
Numero Invertido:	11
Diagrama de Flujo:	12
Números Primos:	12
Números parEimpar:	13
Numero Invertido:	14
Código de lenguaje C:	15
Números primos:	15
Numero parEimpar:	17
Numero Invertido:	19
Conclusión:	21

Introducción:

Es un lenguaje de programación de propósito general, creado por Dennis Ritchie en 1972 en los Laboratorios Bell, diseñado para ser eficiente y versátil.

C es un lenguaje estructurado, lo que significa que los programas se organizan en módulos o funciones, facilitando la lectura y el mantenimiento del código.

Características principales:**Lenguaje de nivel medio:**

C se encuentra entre los lenguajes de bajo nivel (como el ensamblador) y los de alto nivel (como Python o Java). Esto significa que proporciona un buen equilibrio entre la potencia de control sobre el hardware y la facilidad de programación.

Amplia gama de aplicaciones:

C se utiliza para desarrollar desde sistemas operativos y compiladores hasta aplicaciones de usuario, videojuegos y software embebido.

Eficiencia:

Los programas en C suelen ser rápidos y eficientes, lo que los hace adecuados para aplicaciones donde el rendimiento es crítico.

Ventajas de aprender C:

Base para otros lenguajes:

Aprender C proporciona una sólida base para aprender otros lenguajes de programación, como C++, Java, o Python, que incorporan conceptos y estructuras de C.

Comprensión de sistemas:

C es ideal para comprender cómo funcionan los sistemas operativos y el hardware, lo que es valioso para desarrolladores que trabajan a bajo nivel.

Descripción:

Es un lenguaje de programación de propósito general, conocido por su eficiencia y su capacidad para acceder a bajo nivel a la memoria.

C es un lenguaje estructurado y ofrece control sobre el hardware, lo que lo hace adecuado para tareas que requieren alto rendimiento.

Características principales:**Lenguaje compilado:**

El código C se traduce a código máquina antes de la ejecución, lo que resulta en programas rápidos y eficientes.

Estructurado:

C promueve un estilo de programación estructurado, dividiendo el código en funciones y bloques lógicos para mejorar la organización y el mantenimiento.

Soporte de bibliotecas:

C cuenta con una extensa biblioteca estándar que proporciona funciones para tareas comunes como entrada/salida, manipulación de cadenas y matemáticas.

Portabilidad:

Aunque es un lenguaje de bajo nivel, C permite que los programas sean relativamente fáciles de adaptar a diferentes plataformas y sistemas operativos.

Usos comunes:

Sistemas operativos:

C es el lenguaje principal para la implementación de muchos sistemas operativos, incluyendo Unix y sistemas basados en Unix.

Desarrollo de compiladores:

C se utiliza ampliamente en la creación de compiladores para otros lenguajes de programación.

Desarrollo de drivers:

C se utiliza para escribir controladores de dispositivos que interactúan directamente con el hardware.

Justificación:

Eficiencia, portabilidad y control sobre el hardware, lo que lo hace ideal para sistemas operativos, software embebido y aplicaciones que requieren alto rendimiento. Además, su influencia en otros lenguajes y su amplia comunidad de usuarios lo mantienen relevante.

Ventajas del lenguaje C:**Eficiencia y rendimiento:**

C permite un control cercano del hardware, lo que se traduce en programas rápidos y eficientes.

Influencia en otros lenguajes:

C ha sido la base para el desarrollo de muchos otros lenguajes de programación populares, como C++, Java y C#, lo que facilita la transición entre ellos.

Control sobre el hardware:

C permite un acceso directo a la memoria y a los recursos del sistema, lo que es crucial para el desarrollo de sistemas operativos, drivers y software embebido.

Usos comunes del lenguaje C:**Desarrollo de sistemas operativos:**

Muchos sistemas operativos, incluyendo versiones de Unix y Linux, fueron escritos en C.

Desarrollo de bases de datos:

C se utiliza para implementar algunas de las bases de datos más populares a nivel mundial.

Desarrollo de compiladores e intérpretes:

C es un lenguaje adecuado para crear herramientas de desarrollo.

Software embebido:

C es ampliamente utilizado en sistemas embebidos, como microcontroladores y dispositivos IoT, debido a su eficiencia y control sobre el hardware.

Desarrollo:

Algoritmos:

Numero primos:

Algoritmo Primos

```

    Escribir "Dame un numero entero"
    Leer numeroIngresado;
    iteracion = 1;
    Mientras iteracion <= numeroIngresado Hacer
        sí (numeroIngresado % iteracion == 0) Entonces
            divisionResiduoCero = divisionResiduoCero + 1;
        Fin Si
        iteracion=iteracion + 1
    Fin Mientras
    sí(divisionResiduoCero==2) Entonces
        Escribir "Es Primo"
    Sino
        Escribir "No es Primo"
    Fin Si

```

FinAlgoritmo

Imagen de Numero Primos:

The screenshot shows the PSeInt software interface. The menu bar includes Archivo, Editar, Configurar, Ejecutar, and Ayuda. The toolbar contains various icons for file operations, editing, and execution. The workspace displays three files: Primos.psc, parEimpar.psc, and Invertido.psc. The active file, Primos.psc, contains the following Pascal code:

```

1  Algoritmo Primos
2      Escribir "Dame un numero entero"
3      Leer numeroIngresado;
4      iteracion = 1;
5      Mientras iteracion ≤ numeroIngresado Hacer
6          sí (numeroIngresado % iteracion == 0) Entonces
7              divisionResiduoCero = divisionResiduoCero + 1;
8          Fin Si
9          iteracion=iteracion + 1
10 Fin Mientras
11 sí(divisionResiduoCero==2) Entonces
12 Escribir "Es Primo"
13 Sino
14     Escribir "No es Primo"
15 Fin Si
16
17
18 FinAlgoritmo
  
```

On the left side of the workspace, there is a vertical toolbar with icons for variables, operators, and functions, and a list of variables.

Números parEimpar:

Algoritmo parEimpar

num=0;

Repetir

num=num+1;

Leer nro;

sí (nro mod 2) =0 Entonces

Escribir "es par"

Sino

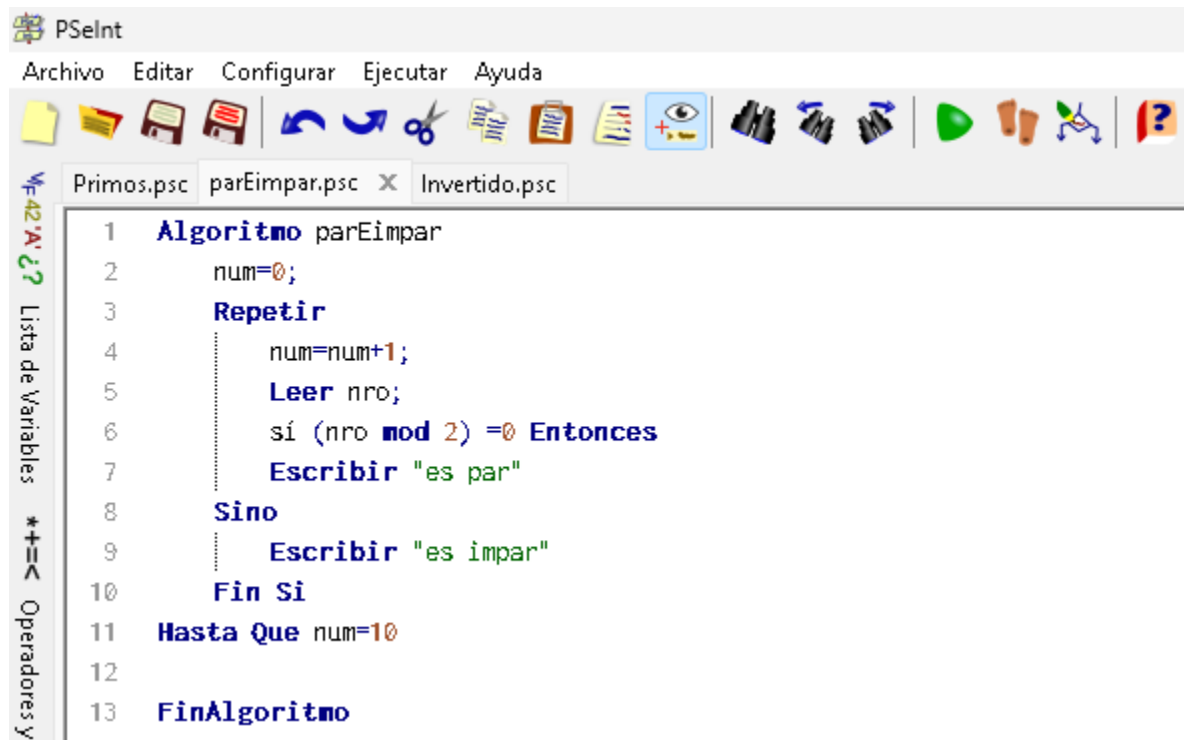
Escribir "es impar"

Fin Si

Hasta Que num=10

FinAlgoritmo

Imagen del Número parEimpar:



The image shows the PSeInt software interface. The menu bar includes Archivo, Editar, Configurar, Ejecutar, and Ayuda. The toolbar contains icons for file operations (new, open, save, print), editing (undo, redo, cut, copy, paste), and execution (run, stop, step-through). The window title bar shows three open files: Primos.psc, parEimpar.psc (active), and Invertido.psc. The active window displays the following pseudocode:

```
1  Algoritmo parEimpar
2      num=0;
3      Repetir
4          .....
4          num=num+1;
5          Leer nro;
6          sí (nro mod 2) =0 Entonces
7              Escribir "es par"
8          Sino
9              Escribir "es impar"
10         Fin Si
11 Hasta Que num=10
12
13 FinAlgoritmo
```

On the left side of the editor, there is a vertical toolbar with icons for variables, lists, and operators, and a label "Lista de Variables y Operadores y".

Numero Invertido:

Algoritmo Invertido

```

    inverso <- 0
    Escribir "Ingresa número"
    Leer numero
    residuo <- numero
    Mientras residuo >0 Hacer
        inverso <- inverso * 10 + residuo mod 10;
        residuo <- (residuo - residuo mod 10) / 10;
    Fin Mientras
    Escribir "valor del número inverso: ", inverso;

```

FinAlgoritmo

Imagen del Número Invertido:

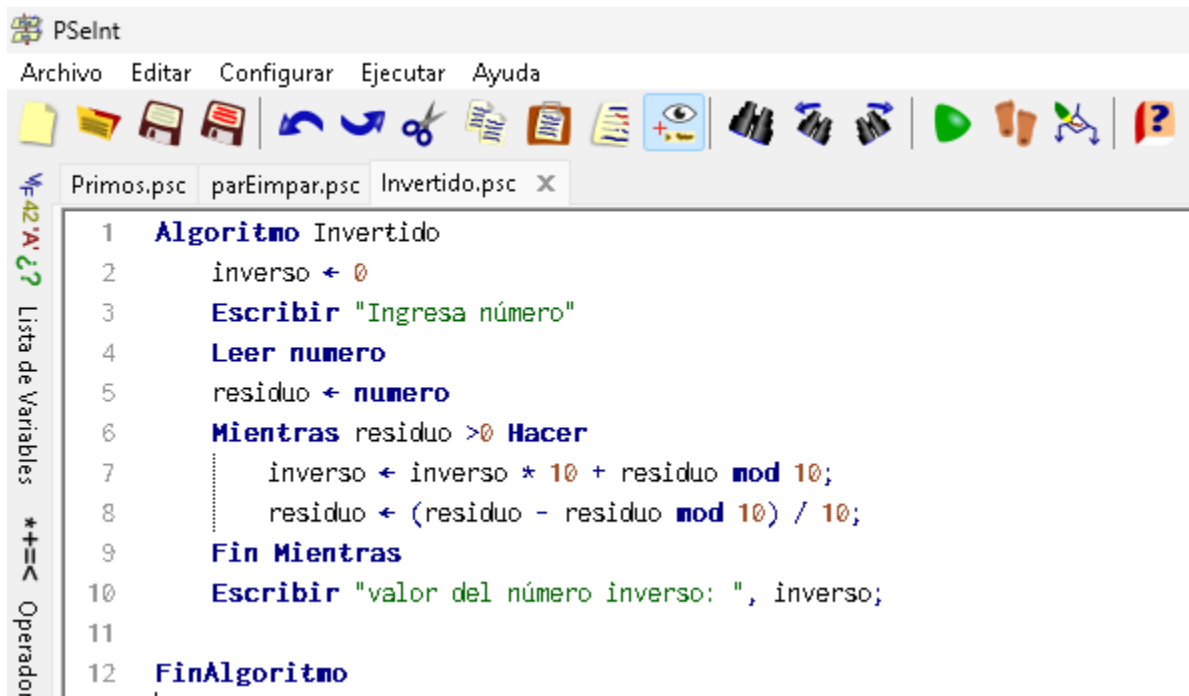
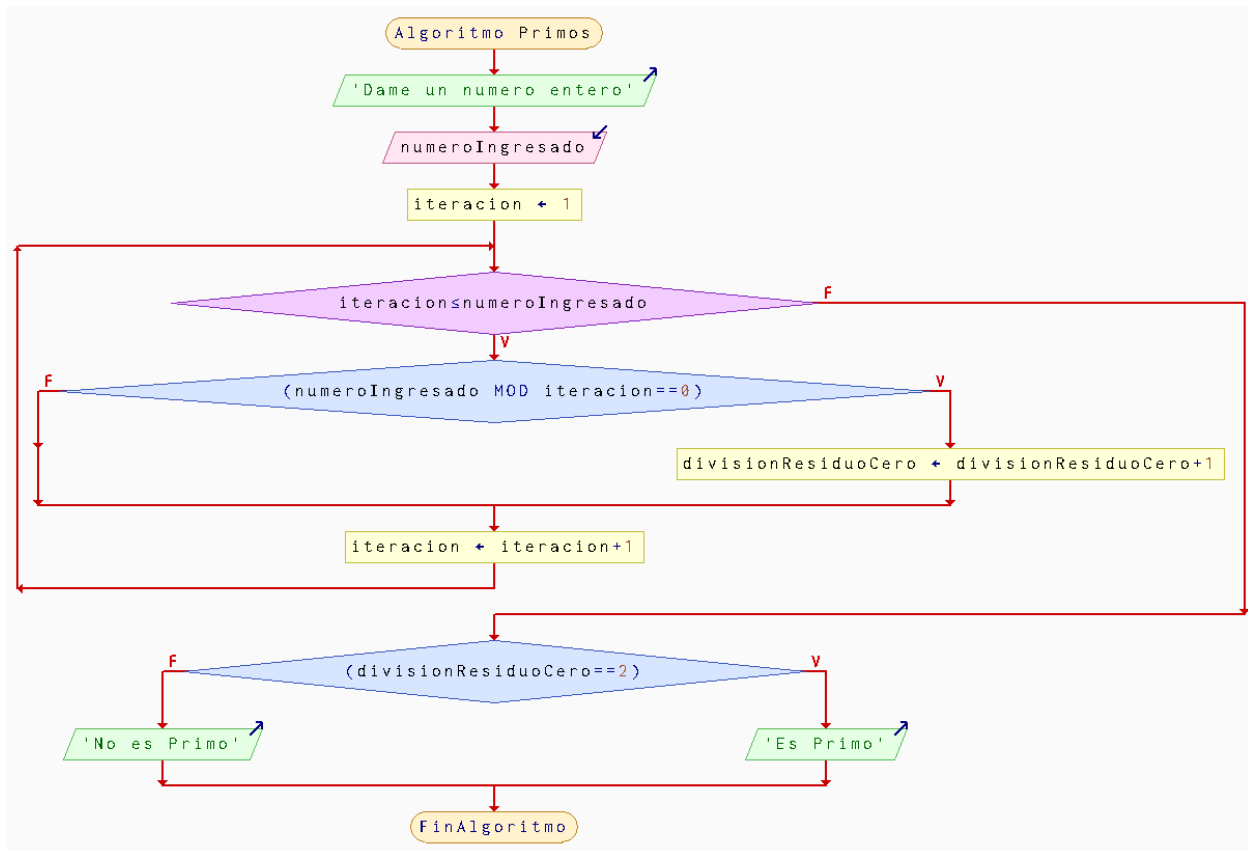
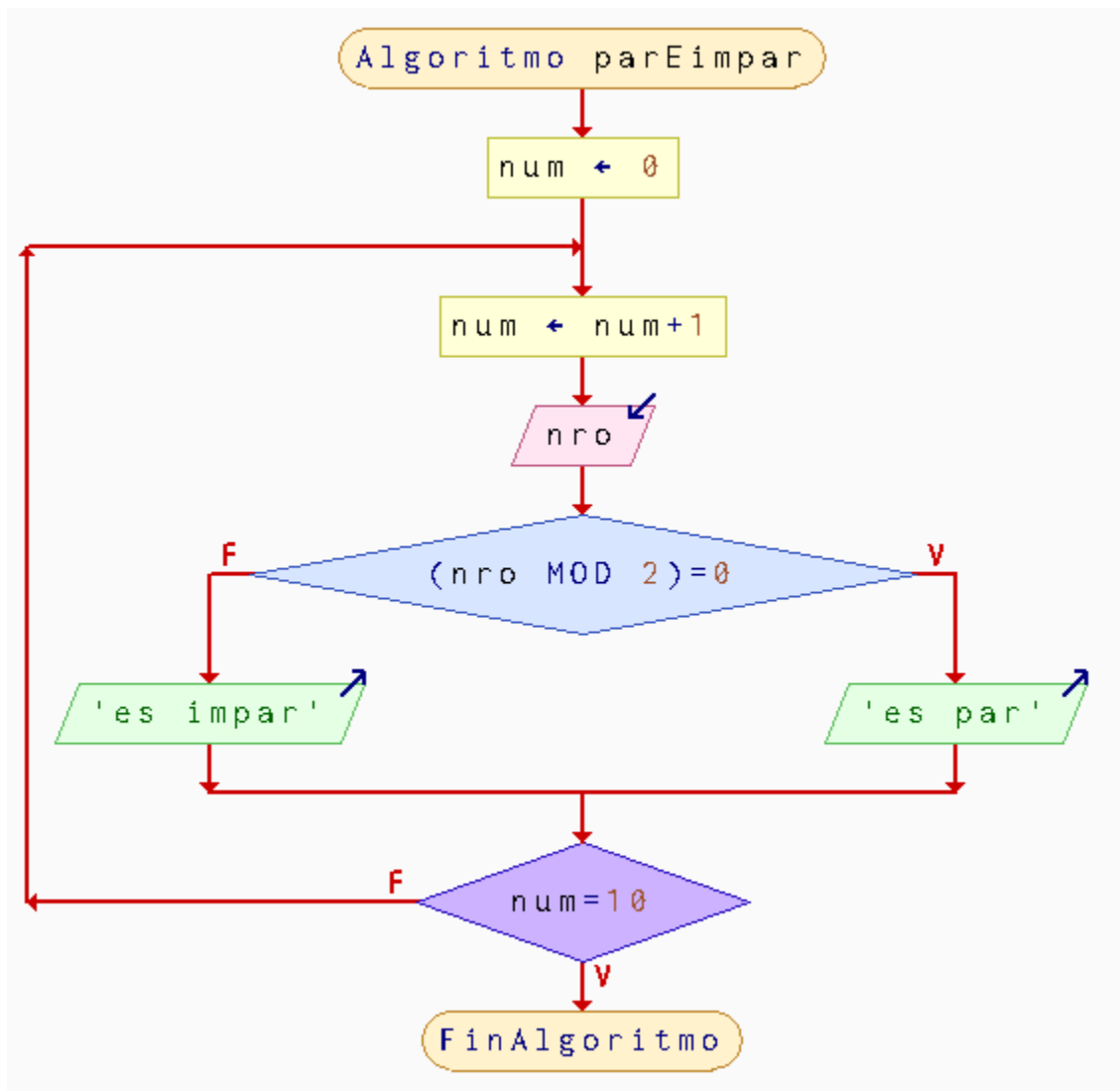
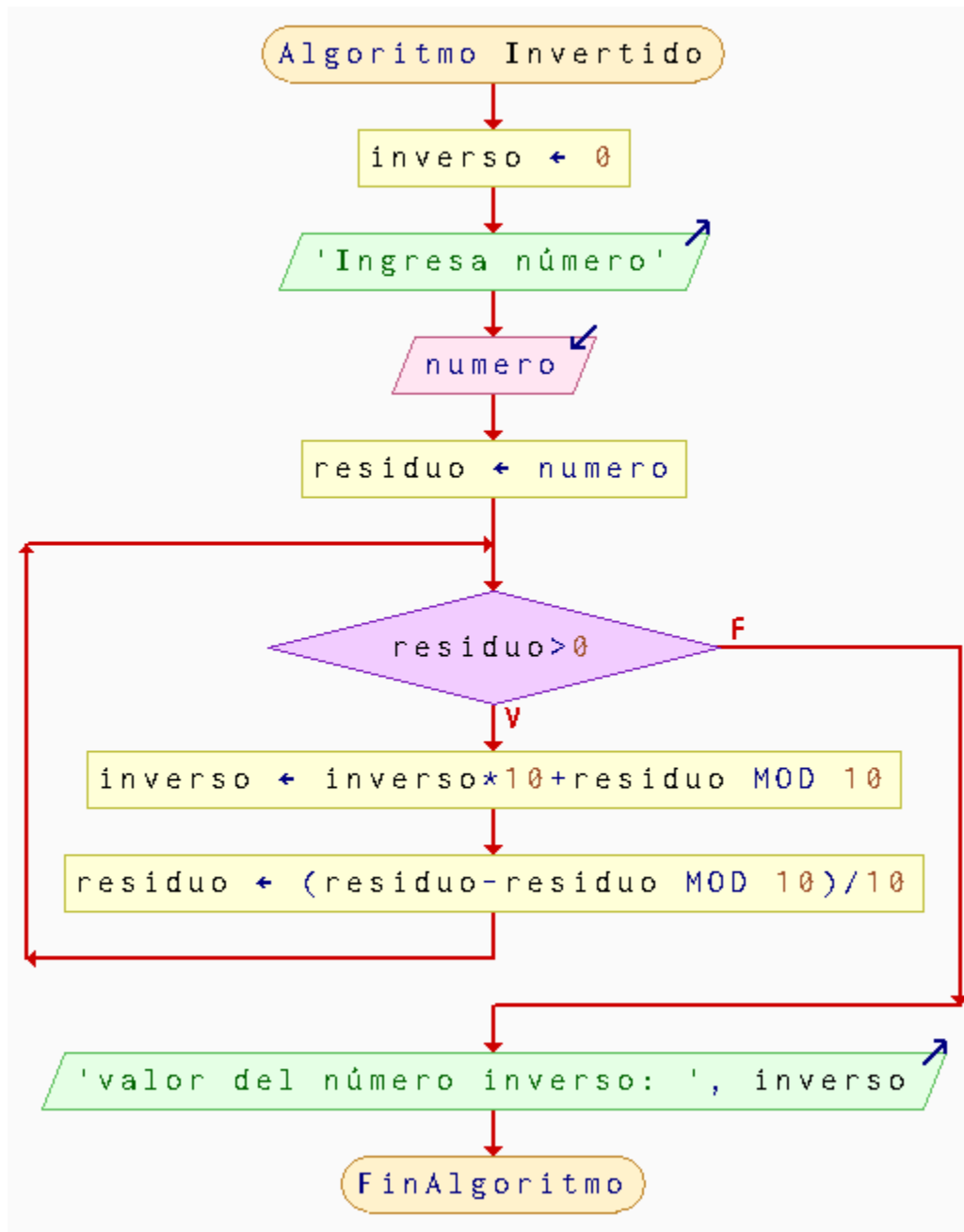


Diagrama de Flujo:

Números Primos:



Números parEimpar:

Numero Invertido:

Código de lenguaje C:**Números primos:**

```
/*Codigo de Numeros Primos*/

#include <stdio.h>
#include <stdbool.h>

int main () {
    int num;
    bool esPrimo = true;

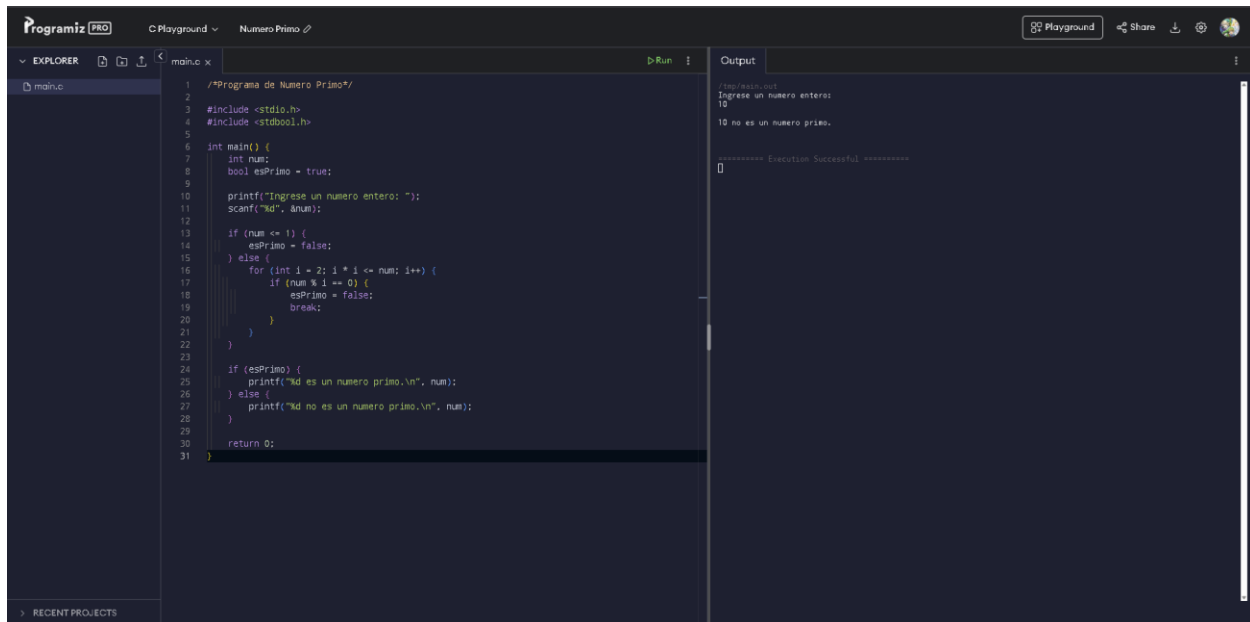
    printf("Ingrese un numero entero: ");
    scanf ("%d", &num);

    if (num <= 1) {
        esPrimo = false;
    } else {
        for (int i = 2; i * i <= num; i++) {
            if (num % i == 0) {
                esPrimo = false;
                break;
            }
        }
    }

    if (esPrimo) {
        printf ("%d es un numero primo.\n", num);
    } else {
        printf ("%d no es un numero primo.\n", num);
    }

    return 0;
}
```

ImagenCodigoLenguajeCdeNumerosPrimos:



```
1 /*Programa de Numero Primo*/
2
3 #include <stdio.h>
4 #include <stdbool.h>
5
6 int main() {
7     int num;
8     bool esPrimo = true;
9
10    printf("Ingrese un numero entero: ");
11    scanf("%d", &num);
12
13    if (num <= 1) {
14        esPrimo = false;
15    } else {
16        for (int i = 2; i * i <= num; i++) {
17            if (num % i == 0) {
18                esPrimo = false;
19                break;
20            }
21        }
22    }
23
24    if (esPrimo) {
25        printf("%d es un numero primo.\n", num);
26    } else {
27        printf("%d no es un numero primo.\n", num);
28    }
29
30    return 0;
31 }
```

Output

```
/*precompilado*/
Ingrese un numero entero:
10
10 no es un numero primo.

Execution Successful
```

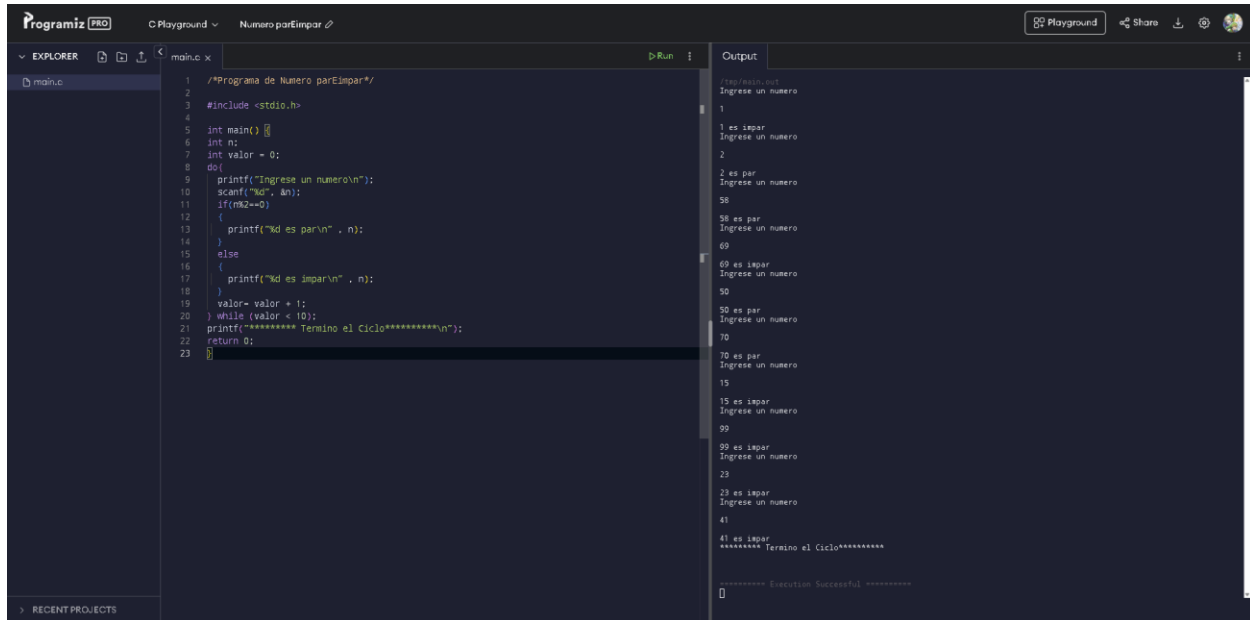
Numeros parEimpar:

```
/*Programa de Numero parEimpar*/

#include <stdio.h>

int main () {
int n;
int valor = 0;
do {
printf ("Ingrese un numero\n");
scanf ("%d", &n);
if(n%2==0)
{
printf ("%d es par\n", n);
}
else
{
printf ("%d es impar\n", n);
}
valor= valor + 1;
} while (valor < 10);
printf ("***** Terminó el Ciclo*****\n");
return 0;
}
```

Imagen Codigo lenguaje C de Numero parEimpar:



```
1  /*Programa de Numero parEimpar*/
2
3  #include <stdio.h>
4
5  int main() {
6      int n;
7      int valor = 0;
8      do{
9          printf("Ingrese un numero\n");
10         scanf("%d", &n);
11         if(n%2==0)
12         {
13             printf("%d es par\n", n);
14         }
15         else
16         {
17             printf("%d es impar\n", n);
18         }
19         valor= valor + 1;
20     } while (valor < 10);
21     printf("***** Terminó el Ciclo*****\n");
22     return 0;
23 }
```

Output

```
Programiz.net
Ingrese un numero
1
1 es impar
Ingrese un numero
2
2 es par
Ingrese un numero
58
58 es par
Ingrese un numero
69
69 es impar
Ingrese un numero
50
50 es par
Ingrese un numero
70
70 es par
Ingrese un numero
15
15 es impar
Ingrese un numero
99
99 es impar
Ingrese un numero
23
23 es impar
Ingrese un numero
41
41 es impar
***** Terminó el Ciclo*****

***** Execution Successful *****
```

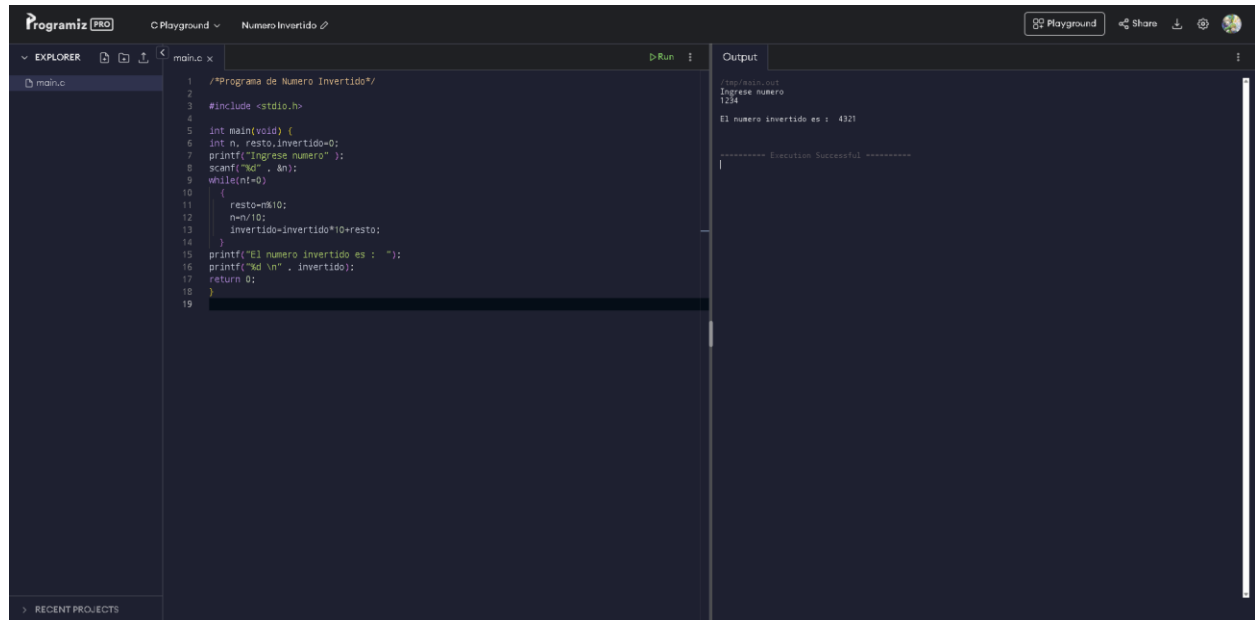
Numero Invertido:

```
/*Programa de Numero Invertido*/
```

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

```
int main(void) {  
    int n, resto, invertido=0;  
    printf ("Ingrese número");  
    scanf ("%d", &n);  
    while (n!=0)  
    {  
        resto=n%10;  
        n=n/10;  
        invertido=invertido*10+resto;  
    }  
    printf ("El numero invertido es: ");  
    printf ("%d \n", invertido);  
    return 0;  
}
```

ImagenCodigo lenguaje C de Numeros Invertido:



The image shows a screenshot of the Programiz Playground interface. The main editor displays a C program designed to reverse the digits of a number. The program uses a loop to extract the last digit of the number and build the reversed number by multiplying the current reversed number by 10 and adding the extracted digit. The input number is 1234, and the output is 4321.

```
1  /*Programa de Numero Invertido*/
2
3  #include <stdio.h>
4
5  int main(void) {
6      int n, resto, invertido=0;
7      printf("Ingrese numero ");
8      scanf("%d", &n);
9      while(n!=0)
10     {
11         resto=n%10;
12         n=n/10;
13         invertido=invertido*10+resto;
14     }
15     printf("El numero Invertido es : ");
16     printf("%d\n", invertido);
17     return 0;
18 }
19
```

The output window shows the execution results:

```
Programiz Playground
Ingrese numero
1234
El numero invertido es : 4321

***** Execution Successful *****
```

Conclusión:

Es un lenguaje fundamental en la historia de la informática, reconocido por su eficiencia, flexibilidad y capacidad de interactuar directamente con el hardware.

Su influencia se extiende a otros lenguajes y tecnologías, sentando las bases para el desarrollo de software moderno.

Puntos clave sobre el código en C:**Eficiencia y rendimiento:**

C permite optimizar el código para lograr un alto rendimiento, especialmente en aplicaciones donde la velocidad y la gestión de recursos son cruciales.

Versatilidad:

C se utiliza en una amplia gama de aplicaciones, desde sistemas operativos y compiladores hasta software embebido y aplicaciones científicas.

Acceso a bajo nivel:

C ofrece acceso directo a la memoria y al hardware, lo que es esencial para tareas como la programación de sistemas operativos y sistemas embebidos.

Necesidad de disciplina:

La programación en C requiere atención a los detalles y comprensión de los conceptos de bajo nivel, pero a cambio ofrece un gran control y flexibilidad.

C sigue siendo un lenguaje relevante y poderoso, apreciado por su eficiencia, su capacidad para interactuar con el hardware y su influencia en la evolución de la programación.

Su aprendizaje proporciona una base sólida para comprender otros lenguajes y tecnologías, y su versatilidad lo mantiene relevante en diversos campos de la informática.

GitHub: <https://github.com/deyvi23456/-Introduccion-Desarrollo-Software-.git>