

Actividad | #3 |

Códigos en Lenguaje C

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Devora

ALUMNO: Oscar Esteban Sánchez Leyva

FECHA: 03/Diciembre/2024

ÍNDICE

ÍNDICE	2
INTRODUCCIÓN	3
DESCRIPCIÓN	4
JUSTIFICACIÓN	5
DESARROLLO	6
NÚMEROS PRIMOS	6
NÚMERO PAR E IMPAR	10
NÚMEROS INVERTIDOS	13
CONCLUSIÓN	16
REFERENCIAS	17

INTRODUCCIÓN

En el presente documento se hablará acerca de los códigos de lenguaje C de programación, que son lenguajes creados por el ser humano para poder comunicarse con las computadoras; así, podríamos decir que el lenguaje de programación es el conjunto de símbolos y palabras que permiten al usuario de una computadora darle instrucciones y órdenes para que la computadora los ejecute.

Los códigos en el lenguaje de programación C se componen de elementos como:

- **Tokens:** Palabras clave, identificadores, constantes, variables o cualquier símbolo que tenga un significado en el lenguaje. La sintaxis de C determina cómo combinar los tokens para escribir el código.
- **Comentarios:** Texto sin formato que no se compila y se usa para explicar partes del programa. Para escribir un comentario de una sola línea se usa `//`, y para escribir un comentario de varias líneas se usa `/* */`.
- **Construcciones de control de flujo:** Toma de decisiones (if-else), selección de una opción de un conjunto de casos posibles (switch), ciclos con criterio de término (while, for), entre otras.
- **Indentación:** Para representar la estructura lógica del programa, se debe indentar el código con tabuladores.
- **Llaves:** Las llaves se deben colocar según el estándar Allman, es decir, en la línea siguiente a un if o a un while.
- **Espacios:** Debe haber un espacio en blanco antes y después de los operadores de comparación, asignación, etc.

DESCRIPCIÓN

El lenguaje C es conocido por su simplicidad y eficiencia. Se usa para escribir código de bajo nivel, lo que es ideal para el desarrollo de sistemas operativos y otros softwares de sistemas.

Los tipos de datos básicos en C son:

- **int:** Define números enteros, con un rango de -32 767 a 32 767.
- **char:** Define caracteres, con un rango de -128 a 127.
- **float:** Define números en punto flotante, con un rango de 3.4E-38 a 3.4E38.
- **double:** Define números en punto flotante de doble precisión, con un rango de 1.7E-308 a 1.7E308.

Características:

- **Estructura:** Los programas en C se organizan en ficheros con extensiones .c y .h.
- **Sintaxis:** La sintaxis de C determina cómo combinar los tokens para escribir el código.
- **Comentarios:** Los comentarios se pueden escribir en una sola línea con // o en varias líneas con /* */.
- **Identificadores:** Los identificadores son los nombres que se les dan a las variables, tipos, funciones y etiquetas del programa.
- **Construcciones de control de flujo:** C incluye construcciones como if-else, switch, while y for.
- **Bibliotecas:** C viene con una biblioteca que contiene funciones para acceder al sistema operativo, entrada y salida formateadas, asignación de memoria y más.

JUSTIFICACIÓN

El código en C se utiliza para diseñar software de sistemas, por lo que se justifica su uso por las siguientes razones:

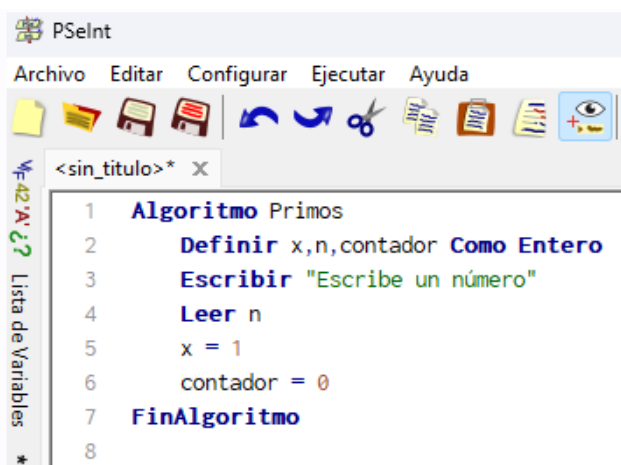
- **Eficiencia:** El código en C es eficiente, lo que lo hace uno de los lenguajes más utilizados para diseñar software de sistemas.
- **Comprender los principios fundamentales de programación:** Estudiar el lenguaje C ayuda a comprender los principios fundamentales de programación y el funcionamiento de los lenguajes en general.
- **Dominar otros lenguajes:** Estudiar C ayuda a aprender otros lenguajes de programación como C++, C#, Java o Python, ya que hay muchas similitudes entre ellos.
- **Creación del sistema operativo Unix:** El lenguaje C se utilizó para crear el sistema operativo Unix, que permitió implementar las minicomputadoras PDP-11.

El lenguaje C es un lenguaje procedimental que se basa en el concepto de funciones. En cambio, el programa en C es una colección de funciones, cada una con un nombre y una lista de argumentos. A su vez, el lenguaje C es una opción popular para el desarrollo de software multiplataforma, ya que el código escrito en C puede ser compilado y ejecutado en diferentes plataformas sin cambios importantes.

DESARROLLO

NÚMEROS PRIMOS

1. “**Definir**” pondremos las variables x, n y contador “**como entero**”. Posteriormente, “**escribir**” pondremos “**escribe un número**” con la función “**leer**” la variable n. Después se va a inicializar con la variable x, a la cual se le va a asignar un 1. También a la variable contador se le va a asignar 0.

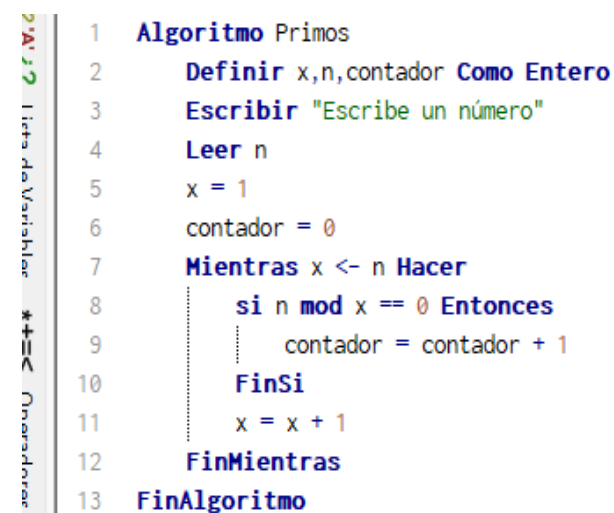


```

1  Algoritmo Primos
2      Definir x,n,contador Como Entero
3      Escribir "Escribe un número"
4      Leer n
5      x = 1
6      contador = 0
7  FinAlgoritmo

```

2. Se va a utilizar un ciclo “**mientras**” que x sea menor o igual a la variable n. “**Hacer**” se va a colocar una condición “**si**” la variable n al dividirla entre lo que vaya valiendo x; si el residuo de la división es igual a 0, “**entonces**” contador es igual a lo que tenga contador más 1. X es igual a lo que tenga x más 1.



```

1  Algoritmo Primos
2      Definir x,n,contador Como Entero
3      Escribir "Escribe un número"
4      Leer n
5      x = 1
6      contador = 0
7      Mientras x <= n Hacer
8          si n mod x == 0 Entonces
9              contador = contador + 1
10         FinSi
11         x = x + 1
12     FinMientras
13 FinAlgoritmo

```

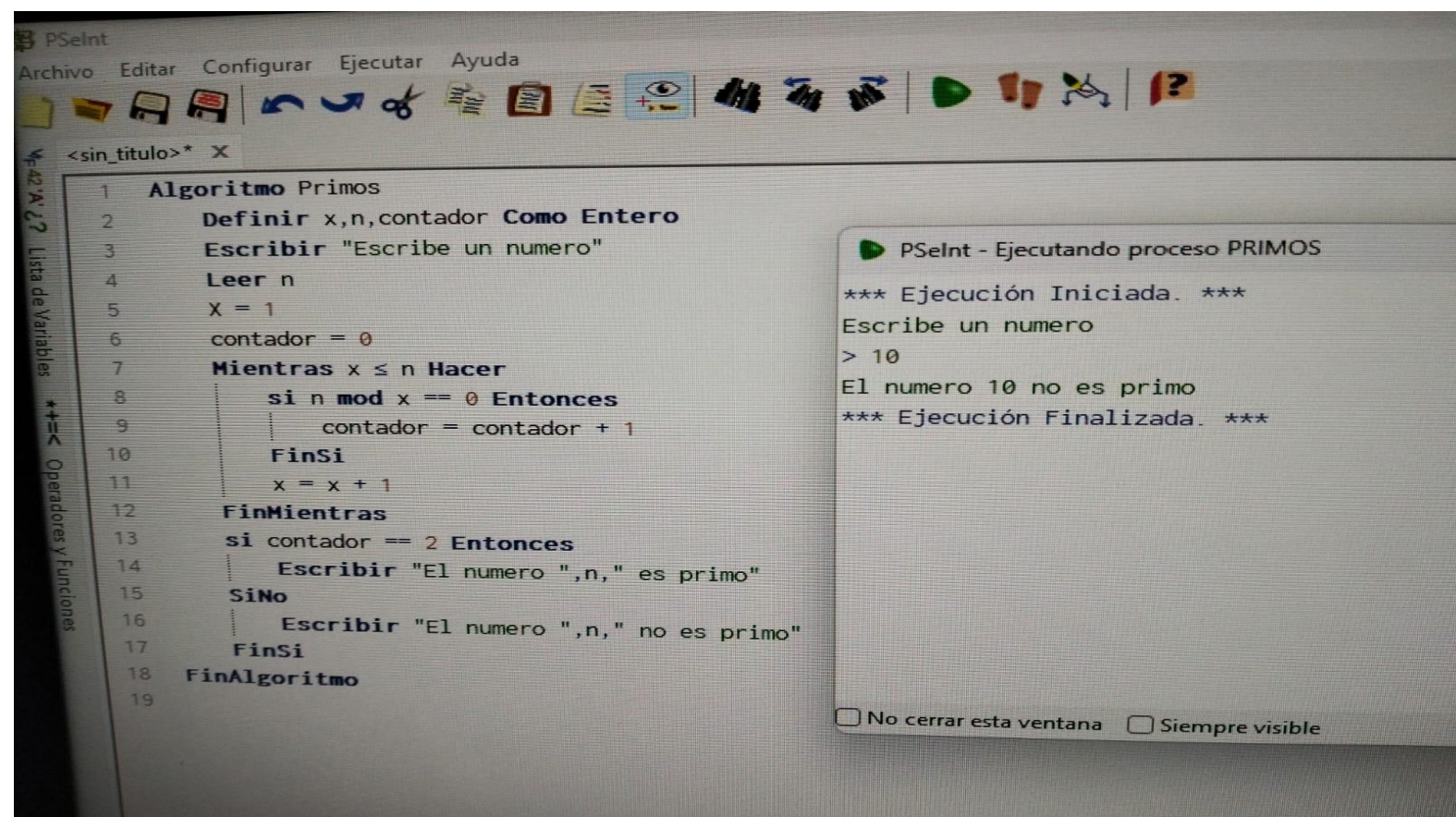
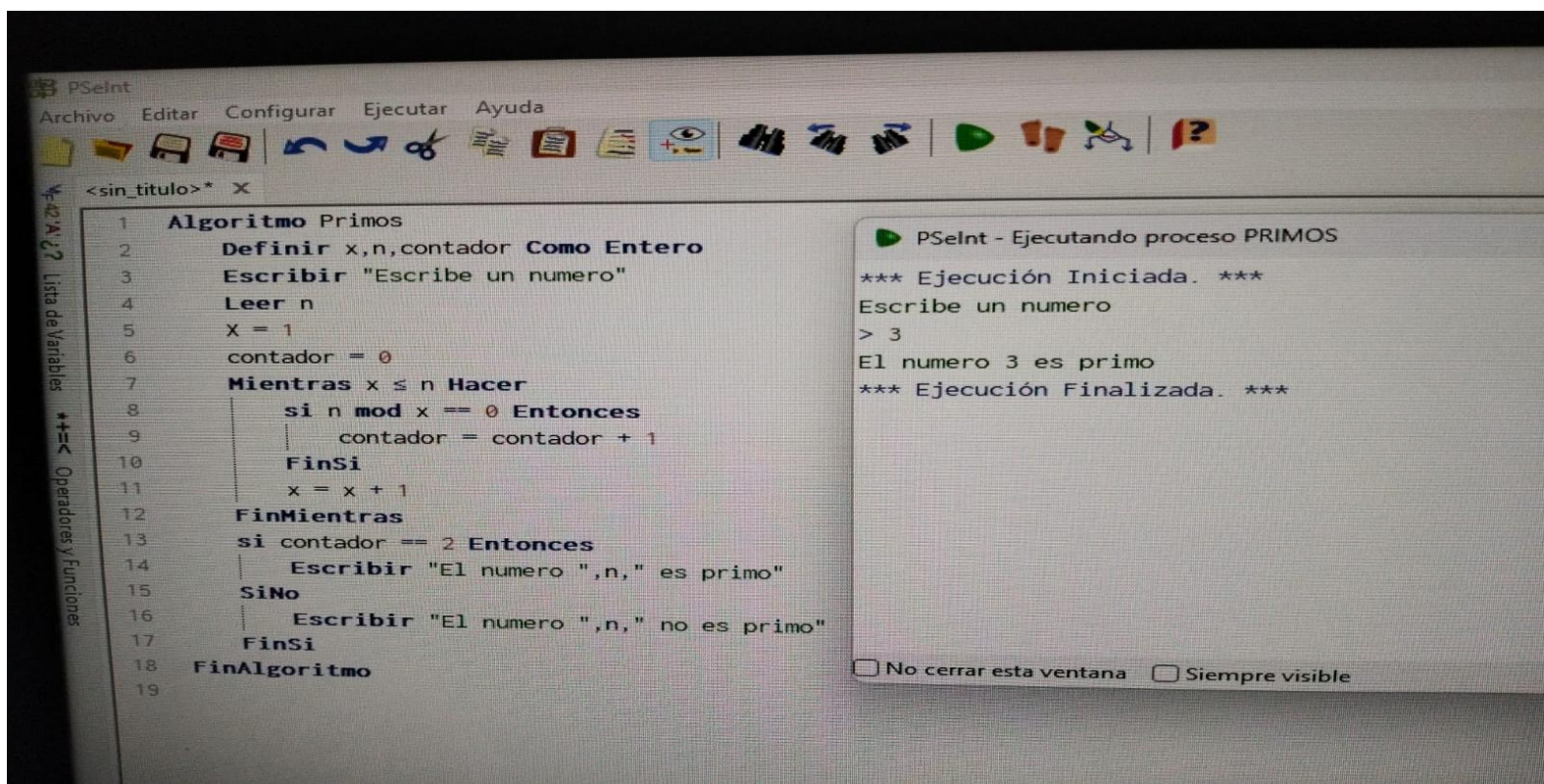
3. Fuera del ciclo, mientras “si” la variable contadora es igual a 2, “entonces” “escribir” “el número” n “es primo”; si esta condición no se cumple, “escribir” “el número” n “no es primo”.

```

1  Algoritmo Primos
2  Definir x,n,contador Como Entero
3  Escribir "Escribe un numero"
4  Leer n
5  x = 1
6  contador = 0
7  Mientras x ≤ n Hacer
8      si n mod x == 0 Entonces
9          contador = contador + 1
10     FinSi
11     x = x + 1
12 FinMientras
13 si contador == 2 Entonces
14     Escribir "El numero ",n," es primo"
15 SiNo
16     Escribir "El numero ",n," no es primo"
17 FinSi
18 FinAlgoritmo

```


4. Ejecutamos.



Para definir un programa matemático en encontrar números primos, puedes usar el algoritmo de la división o la criba de Eratóstenes:

Algoritmo de la división: Para saber si un número es primo, se divide entre 2, 3, 4, 5, ..., $n-1$. Si alguna de las divisiones es exacta, el número es compuesto. Si ninguna de las divisiones es exacta, el número es primo.

Criba de Eratóstenes: Es un algoritmo que permite encontrar muchos números primos menores que un número natural dado. Se atribuye a Eratóstenes de Cirene.

Para buscar nuevos números primos, se puede usar el software Prime95 en el proyecto GIMPS (**Great Internet Mersenne Prime Search**). Este proyecto distribuye la carga de trabajo entre un gran número de ordenadores en todo el mundo.

Para hacer el algoritmo más eficiente, se puede dividir el número entre los números primos menores o iguales a la raíz cuadrada del número. Por ejemplo, para probar si 311 es primo, se puede verificar que no es divisible entre 2, 3, 5, 7, 11, 13 y 17, ya que $\sqrt{311} = 17.635192\dots$

En Python, se puede utilizar el operador módulo % para saber si dos números son divisibles. El operador módulo devuelve el resto de la división entre dos números. Por ejemplo, $5\%3$ es 2.

NÚMERO PAR E IMPAR

1. “**Definir**” las siguientes variables: n, contador uno y contador dos se van a declarar “**como entero**”. “**Escribir**” “**cuántos números deseas contar**”. Se va a asignar con la función “**leer**” en la variable n “**mientras**” la variable n sea mayor al 0. “**Hacer**” n es igual a lo que tenga n menos 1.

```

1  Algoritmo ParesImpares
2      Definir n,contador1,contador2 Como Entero
3      Escribir "Cuantos numeros deseas contar"
4      Leer n
5      Mientras n > 0 Hacer
6          n = n - 1
7      FinMientras
8  FinAlgoritmo
9

```

2. Evaluar “**si**” el número almacenado en la variable n “**mod**” dos es igual a cero; “**entonces**”, contador uno es igual a lo que tenga la variable contador uno más 1. Previamente la voy a inicializar: contador uno es igual a cero; también se va a inicializar la variable contador dos, es igual a cero.

```

1  Algoritmo ParesImpares
2      Definir n,contador1,contador2 Como Entero
3      Escribir "Cuantos numeros deseas contar"
4      Leer n
5      contador1 = 0
6      contador2 = 0
7      Mientras n > 0 Hacer
8          si n mod 2 == 0 Entonces
9              contador1 = contador1 + 1
10         FinSi
11         n = n - 1
12     FinMientras
13 FinAlgoritmo

```

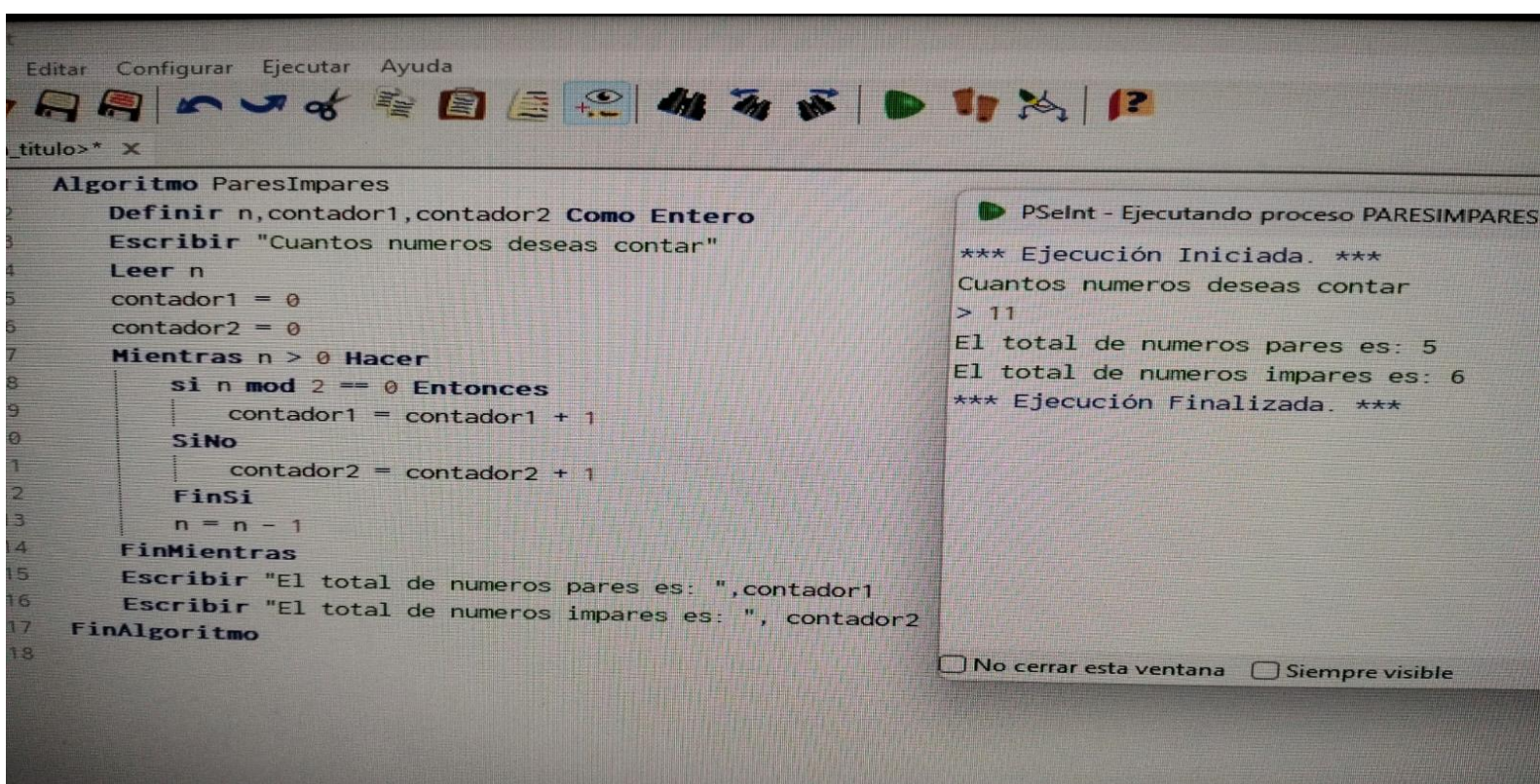
3. Por lo tanto, "sino" contador dos es igual a la variable que tenga contador dos, más uno. Afuera del ciclo, mientras "escribir" "el total de números pares es", concatenamos la variable contador uno; "escribir" "el total de números impares es", concatenamos la variable contador dos.

```

1  Algoritmo ParesImpares
2  Definir n,contador1,contador2 Como Entero
3  Escribir "Cuantos numeros deseas contar"
4  Leer n
5  contador1 = 0
6  contador2 = 0
7  Mientras n > 0 Hacer
8      si n mod 2 == 0 Entonces
9          contador1 = contador1 + 1
10     SiNo
11         contador2 = contador2 + 1
12     FinSi
13     n = n - 1
14 FinMientras
15 Escribir "El total de numeros pares es: ",contador1
16 Escribir "El total de numeros impares es: ", contador2
17 FinAlgoritmo

```

4. Ejecutamos.



Para definir un programa matemático y determinar si un número es par o impar en lenguaje C, se puede utilizar la operación módulo, representada por el símbolo %, y especificar 2 como divisor:

- Si el resultado es 0, el número es par.
- Si el resultado es diferente de 0, el número es impar.

Por ejemplo, para determinar si un número es par en C, se puede utilizar la siguiente instrucción:

Código

```
int numero;  
if (numero % 2 == 0) //Par [3]
```

Para mostrar un mensaje en pantalla, dependiendo si el número es par o impar, se puede utilizar una condición, como la estructura de control IF-ELSE.

O bien para definir un programa matemático que determine si un número es par o impar, se puede utilizar la función Mod. Esta función devuelve el resto de la división de un número entre un divisor. En el caso de los números pares e impares, se debe especificar 2 como divisor. Si el resultado de la división es cero, el número es par, y si no, es impar.

Los números pares son aquellos que se pueden dividir por 2 sin dejar residuo, mientras que los números impares dejan un residuo de 1 al dividirse por 2.

También se puede reconocer si un número es par o impar observando su último dígito. Si la base de numeración es un número par, como la base 10 o la base 8, un número es par si su último dígito también es par. Por ejemplo, en base 10, el número 6 es par porque su último dígito es 6.

NÚMEROS INVERTIDOS

1. Vamos a pedirle al usuario que ingrese un número. “Escribir” “ingrese un número”
se va a asignar con la función “leer” en la variable n, inicializar “definir” n “como entero”
“mientras” que n sea mayor que cero “hacer” n “mod” 10; se guardará en una variable dig.

```

1  Algoritmo Invertidos
2      Definir n Como Entero
3      Escribir "ingrese un numero"
4      Leer n
5      Mientras n>0 Hacer
6          dig←n mod 10
7      FinMientras
8  FinAlgoritmo
9

```

2. Colocamos la variable n “trunc” n entre 10. Inicializar i entre paréntesis, la variable i por 10 más dig.

```

1  Algoritmo Invertidos
2      Definir n Como Entero
3      Escribir "ingrese un numero"
4      Leer n
5      Mientras n>0 Hacer
6          dig←n mod 10
7          i← (i*10) + dig
8          n← trunc(n/10)
9      FinMientras
10 FinAlgoritmo

```

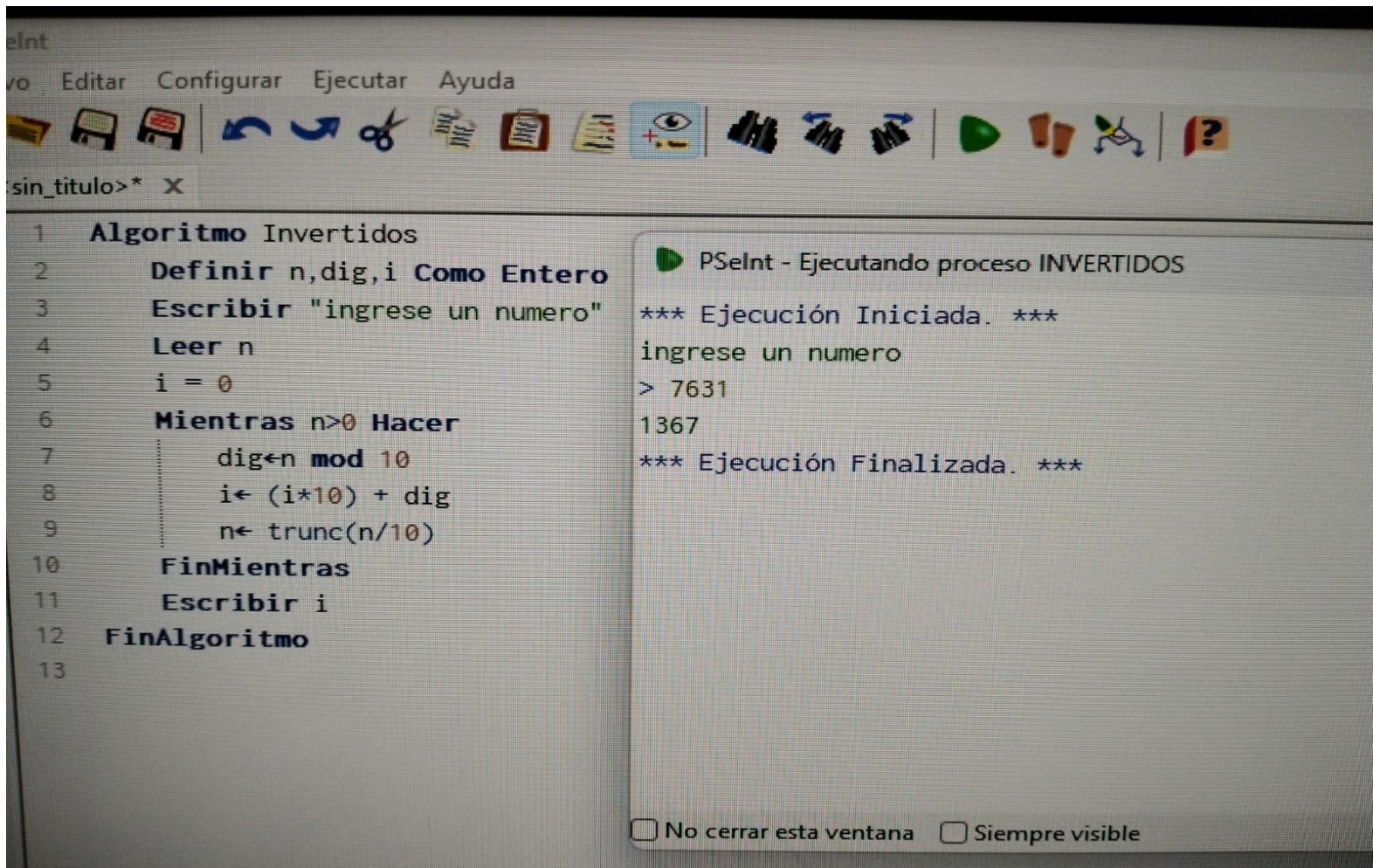

3. Inicializar con la variable i es igual a cero; afuera del ciclo, mientras “**escribir**” la variable i y se inicializa de nuevo colocando dig y la variable i en el segundo renglón.

```

1  Algoritmo Invertidos
2      Definir n,dig,i Como Entero
3      Escribir "ingrese un numero"
4      Leer n
5       $i = 0$ 
6      Mientras  $n > 0$  Hacer
7           $dig \leftarrow n \bmod 10$ 
8           $i \leftarrow (i * 10) + dig$ 
9           $n \leftarrow \text{trunc}(n / 10)$ 
10     FinMientras
11     Escribir i
12 FinAlgoritmo

```

4. Ejecutamos.



Un algoritmo matemático es un conjunto de instrucciones ordenadas que se siguen para resolver un problema. En el caso de un algoritmo para invertir un número, el programa matemático se basa en dividir el número entre 10 y tomar el resto para extraer los dígitos más bajos.

En criptografía, el inverso de un número A es $1/A$, ya que $A * 1/A = 1$. Por ejemplo, el inverso de 5 es $1/5$. Todos los números reales que no sean 0 tienen un inverso.

Los números reales que no sean 0 tienen un inverso. Multiplicar un número por el inverso de A es lo mismo que dividir entre A . Por ejemplo, $10/5$ es lo mismo que $10 * 1/5$.

Los algoritmos se construyen con tres bloques básicos: secuenciación, selección e iteración. El orden de los pasos es crucial para que el algoritmo sea correcto.

Descripción

Proceso	Se divide el número entre 10 y se toma el resto, que corresponde al último dígito del número original.
Construcción	Se multiplica el valor actual del número invertido por 10 y se le suma el resto calculado.
Repetición	Se repite el proceso hasta que el número sea menor o igual a 0.
Resultado	El número invertido es la secuencia de restos producida.

CONCLUSIÓN

En general, el lenguaje de programación C es una herramienta fundamental en la informática, ya que es eficiente, flexible y sencillo. Su código es claro y estructurado, lo que facilita la creación de aplicaciones de forma rápida y potente.

Algunas conclusiones sobre el lenguaje de programación C son:

- Es una herramienta fundamental en el mundo de la informática.
- Es ideal para una amplia variedad de aplicaciones, como el desarrollo de sistemas operativos, software de aplicación y dispositivos embebidos.
- Es un lenguaje de nivel medio bajo, por lo que no requiere de muchos recursos.
- Ofrece una buena respuesta y una ejecución rápida de programas.
- Es preferible escribir el código de un microcontrolador en C, ya que se caracteriza por la capacidad de manipular direcciones arbitrarias.
- Programar en C proporciona una base sólida, facilitando el aprendizaje de otros lenguajes como C++, Objective-C, C#, PHP o JavaScript.

Tomando en cuenta que el pseudocódigo es una forma de escribir algoritmos utilizando un lenguaje cercano al natural, lo que facilita la comprensión y el diseño de programas.

REFERENCIAS

Saavedra, J. A. (2023, 1 junio). *Qué es lenguaje C: las ventajas, las características y la sintaxis*. Ebac. <https://ebac.mx/blog/que-es-lenguaje-c>

Trini, S. (2021, 5 mayo). Tipos de datos en C. *la35.net*. <https://la35.net/c/c-data-types.html>

Gomila, J. G. (2024, 19 noviembre). Lenguaje C, Qué es y Cuáles son sus Características. *Frogames*. <https://cursos.frogamesformacion.com/pages/blog/lenguaje-c-que-es>

IBM MQ 9.2. (s. f.). <https://www.ibm.com/docs/es/ibm-mq/9.2?topic=examples-c-language>

ALGORITMODETAREA. (2020, 24 mayo). *Algoritmo para determinar si un número es primo pseint* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=dH07OnMsCCM>

ALGORITMODETAREA. (2021a, febrero 19). *Algoritmo para contar números pares e impares* [Vídeo]. YouTube. https://www.youtube.com/watch?v=qqbrsY4_dj4

Dailymotion. (s. f.). <https://www.dailymotion.com/video/x8n19b1>