

## Actividad | #1 |

### Algoritmos

## Introducción al Desarrollo de Software

### Ingeniería en Desarrollo de Software

---



TUTOR: Sandra Luz Lara Devora

ALUMNO: Oscar Esteban Sánchez Leyva

FECHA: 30/Noviembre/2024

## ÍNDICE

ÍNDICE .....	2
INTRODUCCIÓN .....	3
DESCRIPCIÓN .....	4
JUSTIFICACIÓN .....	5
DESARROLLO .....	6
NÚMEROS PRIMOS .....	6
NÚMEROS PAR E IMPAR .....	13
NÚMEROS INVERTIDOS .....	16
CONCLUSIÓN .....	19
REFERENCIAS .....	20

## INTRODUCCIÓN

En el presente documento se hablará acerca de los algoritmos; para ellos debemos saber que un algoritmo es un conjunto finito de instrucciones o pasos organizados que describe el proceso que se debe seguir para dar solución a un problema específico o también podría ser una secuencia finita de operaciones realizables, no ambiguas, cuya ejecución da una solución de un problema:

- Son una secuencia de pasos que se deben seguir para alcanzar un objetivo.
- Están ordenados y acotados.
- Tienen una entrada, conocida como input, y una salida, conocida como output.
- Se escriben en un lenguaje de programación que el ordenador pueda entender.

Los algoritmos son esenciales en el mundo de la programación, ya que son la base de los programas y sistemas operativos.

Para resolver un problema, se pueden seguir los siguientes pasos:

- Análisis del problema: definir y comprender el problema.
- Diseño del algoritmo: elaborar una secuencia lógica de pasos que resuelva el problema.
- Resolución del algoritmo: expresar el algoritmo como un programa en un lenguaje de programación.

## DESCRIPCIÓN

Si bien los algoritmos generalmente se asocian al ámbito matemático, no necesariamente implica que sean exclusivos de esta área. Los algoritmos se pueden representar como un diagrama de flujo, un lenguaje de programación o un modelo matemático.

Un ejemplo de algoritmo es el de encriptación, que se utiliza en la seguridad informática para proteger la información.

Los algoritmos se pueden encontrar en muchos ámbitos, como:

- La cocina, como una receta.
- La tecnología, como la operación que desbloquea un electrodoméstico.
- Internet, como la predicción del tiempo o las noticias en Facebook.
- La búsqueda en Google, como el algoritmo PageRank.
- Las redes sociales, como el algoritmo Edgerank de Facebook.
- Las plataformas de streaming, como el algoritmo de Netflix.

En resumen, un algoritmo es un conjunto de reglas definidas que permite solucionar un problema, de una determinada manera, mediante operaciones sistemáticas (no necesariamente ordenadas) y finitas. Estas instrucciones, definidas y ordenadas en función de los datos, resuelven el problema o la tarea.

## JUSTIFICACIÓN

La justificación de los algoritmos es importante para aclarar el pensamiento y razonamiento detrás de su diseño, lo que facilita su comprensión, mantenimiento y reutilización. También puede ayudar a demostrar la competencia y credibilidad del diseñador de algoritmos.

Son procesos matemáticos que permiten agilizar la programación, ya que permiten resolver un problema antes de codificarlo.

Su finalidad es seguir una serie de instrucciones (como en una receta). Simulan los procesos de la inteligencia humana a través del aprendizaje, el razonamiento y la autocorrección.

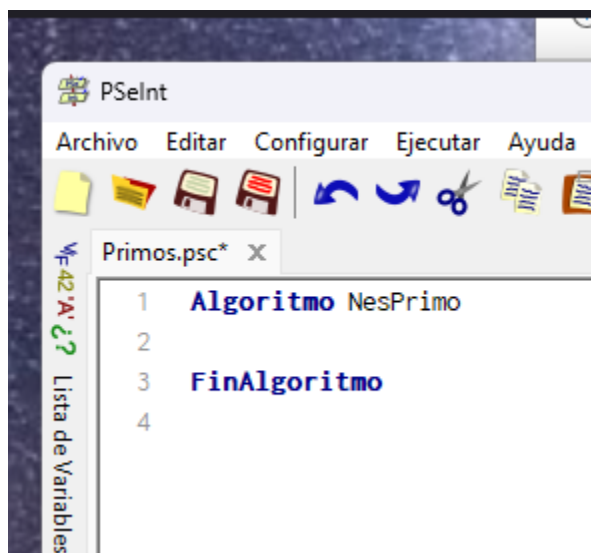
Algunos aspectos a considerar sobre los algoritmos son:

- Los algoritmos se han vuelto muy importantes en la vida cotidiana, estando presentes en los dispositivos electrónicos.
- Existen varios tipos de algoritmos, como los computacionales, no computacionales, cuantitativos y cualitativos.
- El mejor algoritmo para resolver un problema es el que lo resuelve más rápido.
- El estudio teórico del desempeño de los algoritmos y los recursos utilizados por estos es importante para evaluar su rendimiento.

## DESARROLLO

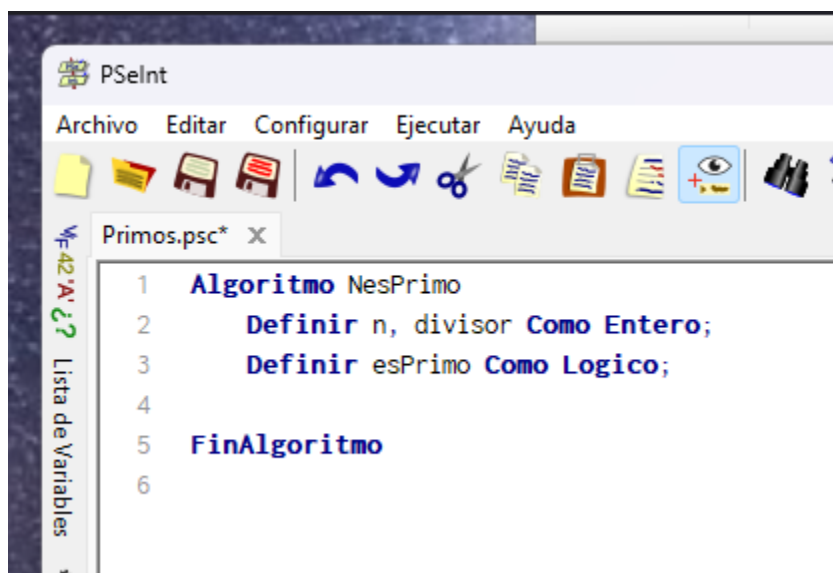
### NÚMEROS PRIMOS

1.- Ingrese a PSeInt, guardar como y crear la carpeta Primos.



2.- Vamos a "definir" una variable, n y el divisor, ambos van a hacer "como enteros".

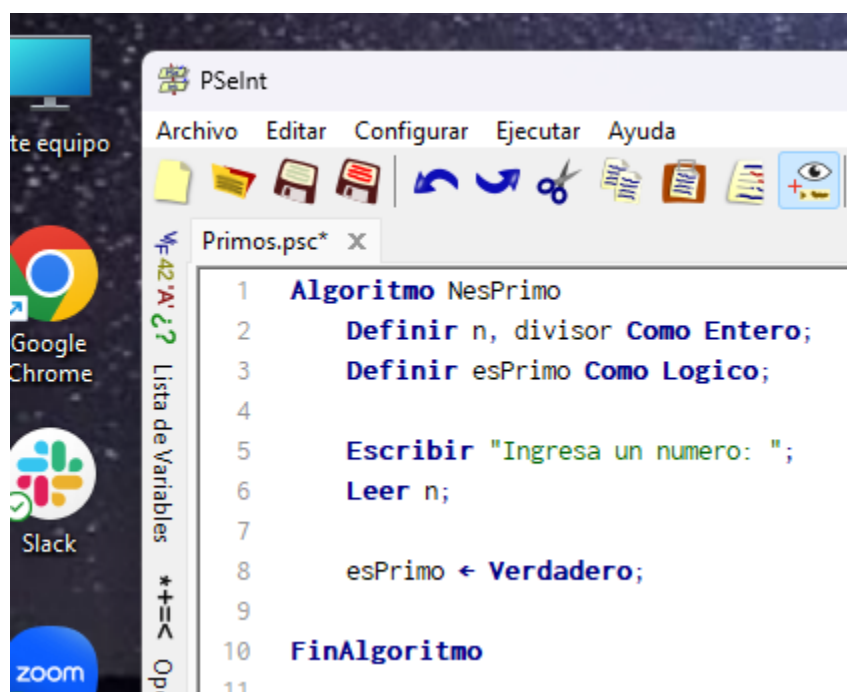
Posteriormente vamos a "definir" una variable "como lógica" que se va a llamar es primo.



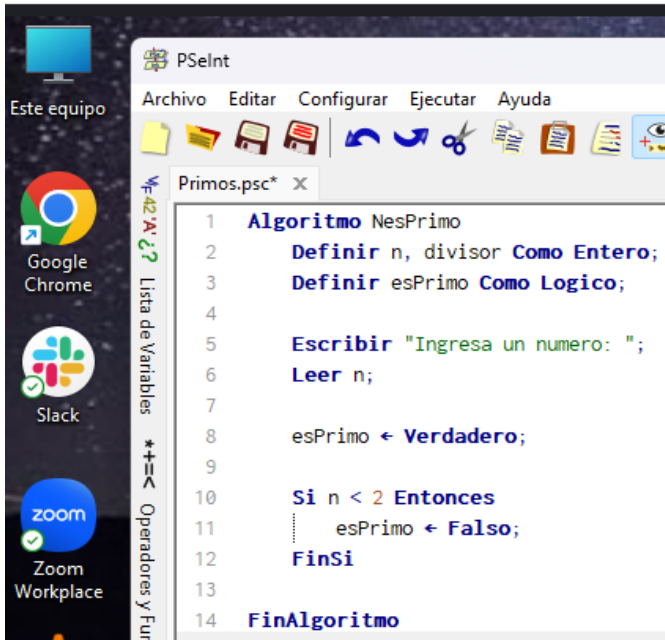
3.- Lo que se va a "escribir" es pedirle al usuario que "ingrese un número". Vamos a "leer" la respuesta del usuario y lo vamos a almacenar en nuestra variable n.



4.- Vamos a suponer que nuestro número sí es primo, se le va a asignar un "verdadero" y se debe hacer una comprobación. El primer número primo es el 2.



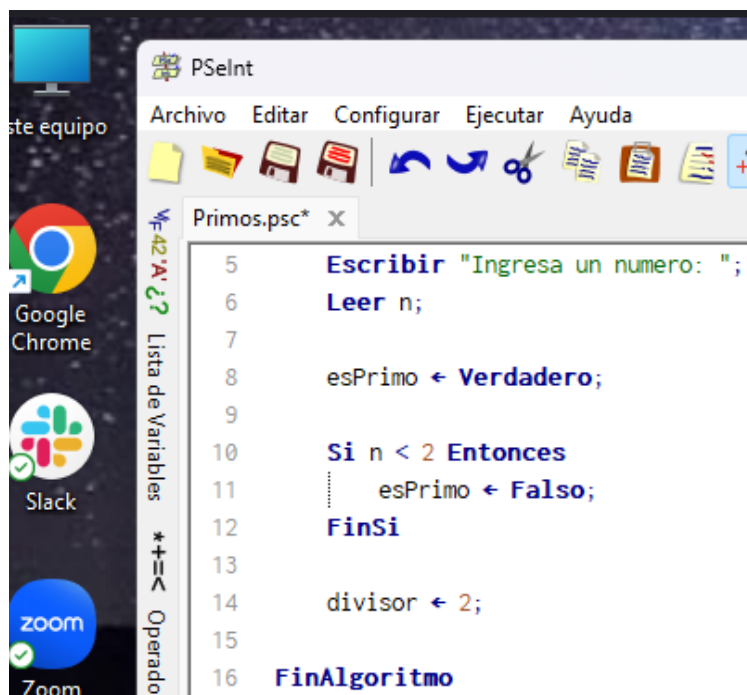
5.- "Si" el número que ingresa al usuario es menor a 2 "entonces" es primo. Va a ser "falso" porque no hay números primos menores a 2.



```

1  Algoritmo NesPrimo
2  Definir n, divisor Como Entero;
3  Definir esPrimo Como Logico;
4
5  Escribir "Ingresa un numero: ";
6  Leer n;
7
8  esPrimo ← Verdadero;
9
10 Si n < 2 Entonces
11     esPrimo ← Falso;
12 FinSi
13
14 FinAlgoritmo
  
```

6.- A la variable divisora se le va a asignar el valor de 2 porque se van a dividir todos los números desde dos hasta uno, menos del número que se tiene.

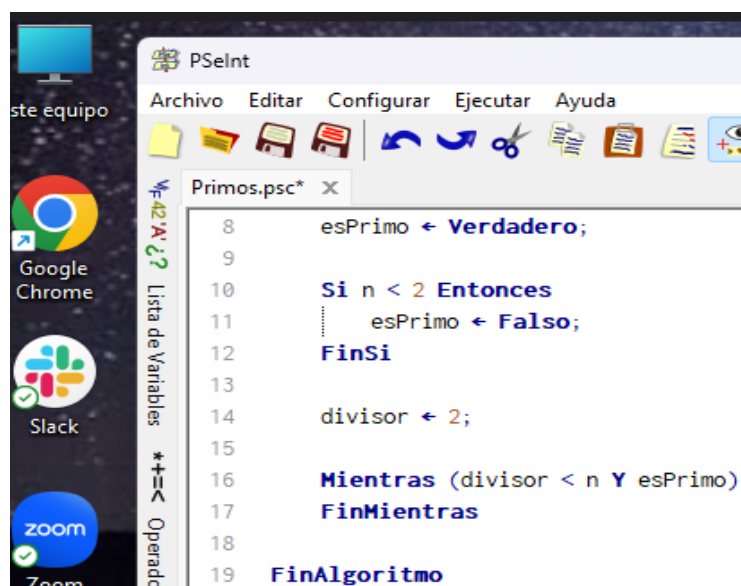


```

5  Escribir "Ingresa un numero: ";
6  Leer n;
7
8  esPrimo ← Verdadero;
9
10 Si n < 2 Entonces
11     esPrimo ← Falso;
12 FinSi
13
14 divisor ← 2;
15
16 FinAlgoritmo
  
```



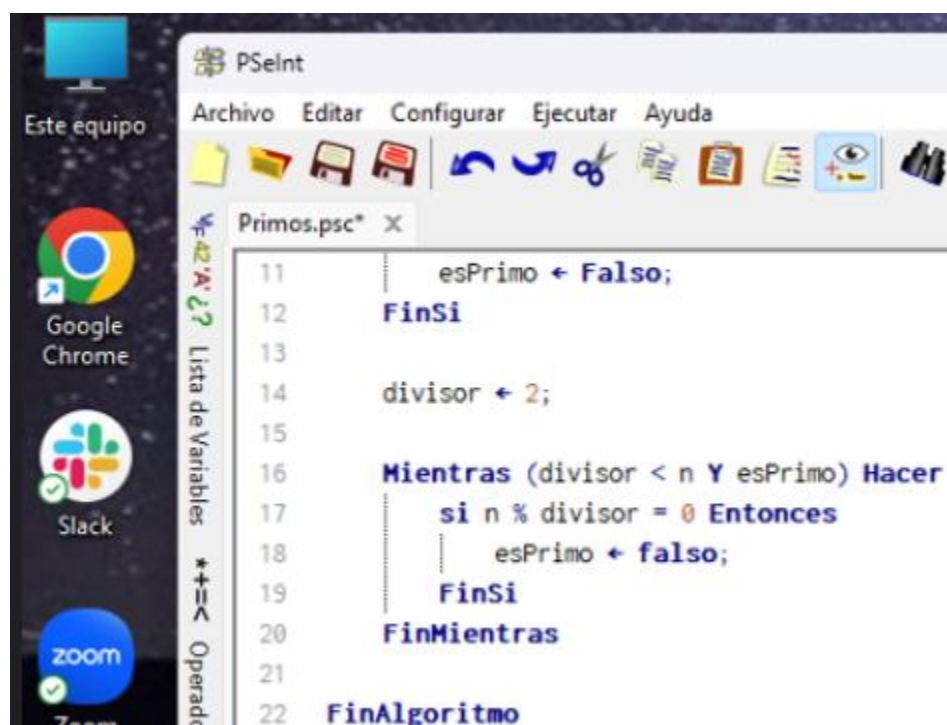
7.- Se va a necesitar dentro de un ciclo "mientras" el divisor sea menor a nuestra variable  $n$ , donde se pondrá otra condición que usará al operador "y" porque las dos condiciones se tienen que cumplir.



```

PSeInt
Archivo  Editar  Configurar  Ejecutar  Ayuda
Primos.psc* X
8      esPrimo ← Verdadero;
9
10     Si n < 2 Entonces
11         esPrimo ← Falso;
12     FinSi
13
14     divisor ← 2;
15
16     Mientras (divisor < n Y esPrimo)
17     FinMientras
18
19     FinAlgoritmo
  
```

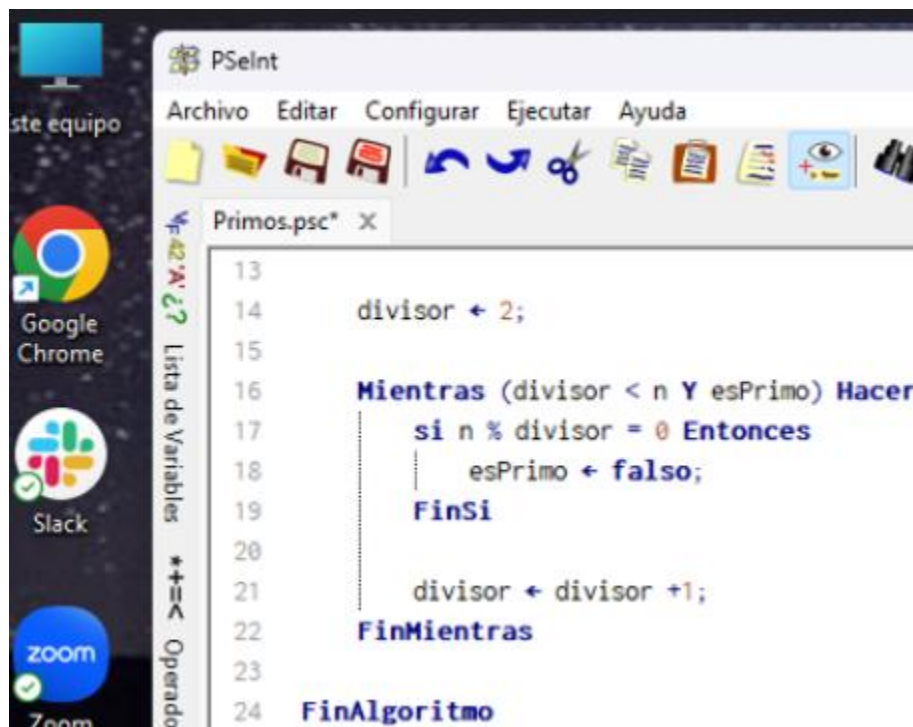
8.- Pondremos "hacer" y dentro del ciclo, mientras, pondremos un "si". Si el módulo divisor es igual a cero, significa que encontramos un divisor adicional de número  $n$ ; la variable es primo va a cambiar su estado a "falso".



```

PSeInt
Archivo  Editar  Configurar  Ejecutar  Ayuda
Primos.psc* X
11         esPrimo ← Falso;
12     FinSi
13
14     divisor ← 2;
15
16     Mientras (divisor < n Y esPrimo) Hacer
17         si n % divisor = 0 Entonces
18             esPrimo ← falso;
19         FinSi
20     FinMientras
21
22     FinAlgoritmo
  
```

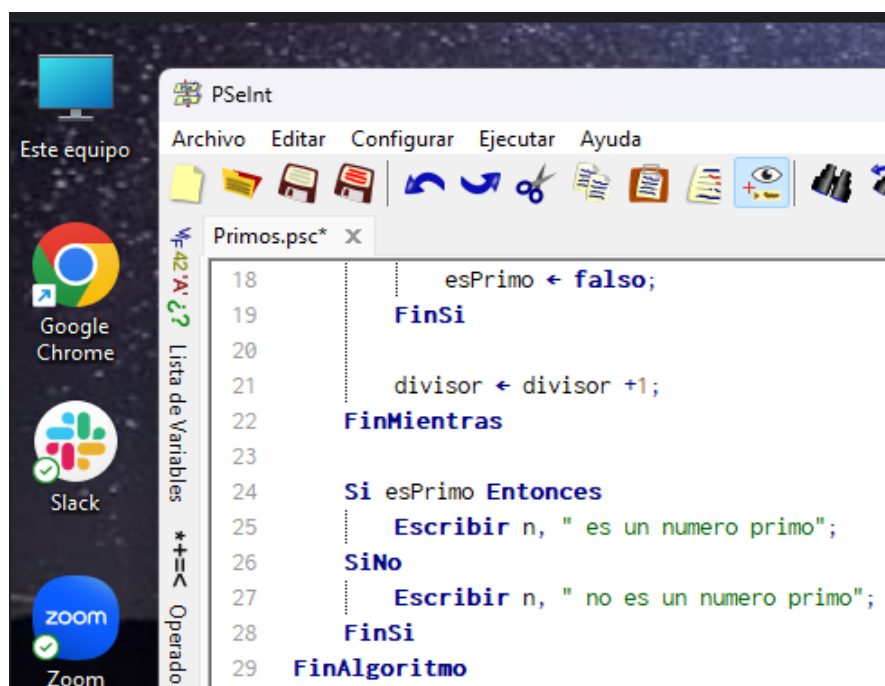
9.- Después de esta condición vamos a aumentar el divisor asignándole lo que ya teníamos, divisor más 1, con eso tenemos todo el ciclo completo.



```

13
14     divisor ← 2;
15
16     Mientras (divisor < n Y esPrimo) Hacer
17     |     si n % divisor = 0 Entonces
18     |     |     esPrimo ← falso;
19     |     FinSi
20     |
21     |     divisor ← divisor +1;
22     FinMientras
23
24     FinAlgoritmo
  
```

10.- Para saber si un número es primo o no, tenemos que mostrar los mensajes correspondientes: "si" nuestra variable es primo, es verdadera, "entonces" pondremos n y después concatenamos el mensaje "es un número primo" "sino" "no es un número primo".



```

18     |     esPrimo ← falso;
19     |     FinSi
20
21     |     divisor ← divisor +1;
22     FinMientras
23
24     Si esPrimo Entonces
25     |     Escribir n, " es un numero primo";
26     SiNo
27     |     Escribir n, " no es un numero primo";
28     FinSi
29     FinAlgoritmo
  
```

The screenshot 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, step-through, stop). The active window is 'Primos.psc'. On the left sidebar, there are icons for a variable list, a help/question mark, and a list of operators and functions. The main editor displays the following Pascal code:

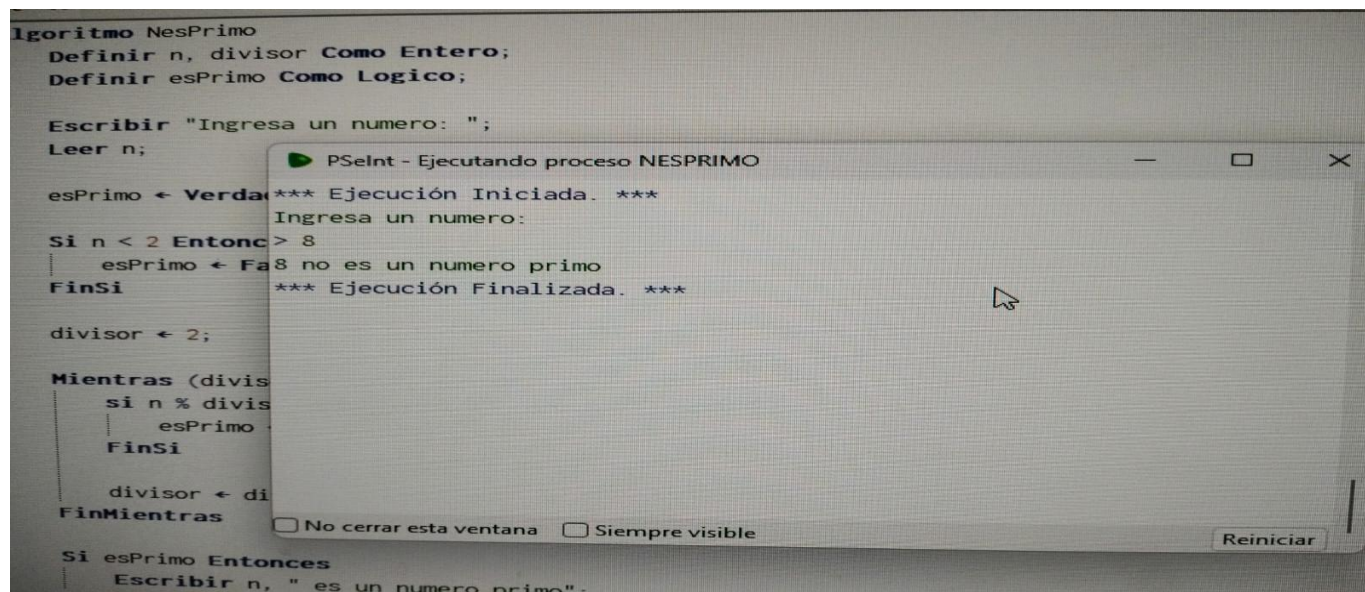
```

1  Algoritmo NesPrimo
2      Definir n, divisor Como Entero;
3      Definir esPrimo Como Logico;
4
5      Escribir "Ingresa un numero: ";
6      Leer n;
7
8      esPrimo  $\leftarrow$  Verdadero;
9
10     Si n < 2 Entonces
11         esPrimo  $\leftarrow$  Falso;
12     FinSi
13
14     divisor  $\leftarrow$  2;
15
16     Mientras (divisor < n Y esPrimo) Hacer
17         si n % divisor = 0 Entonces
18             esPrimo  $\leftarrow$  falso;
19         FinSi
20
21         divisor  $\leftarrow$  divisor + 1;
22     FinMientras
23
24     Si esPrimo Entonces
25         Escribir n, " es un numero primo";
26     SiNo
27         Escribir n, " no es un numero primo";
28     FinSi
29 FinAlgoritmo
30

```



11.- Lo ejecutaremos y lo probaremos con un número par que es el 8. Es divisible entre 2 y el 4, así que nos tendría que decir que no es un número primo.



```

Algoritmo NesPrimo
  Definir n, divisor Como Entero;
  Definir esPrimo Como Logico;

  Escribir "Ingresar un numero: ";
  Leer n;

  esPrimo ← Verdadero;
  Si n < 2 Entonces
    esPrimo ← Falso;
  FinSi

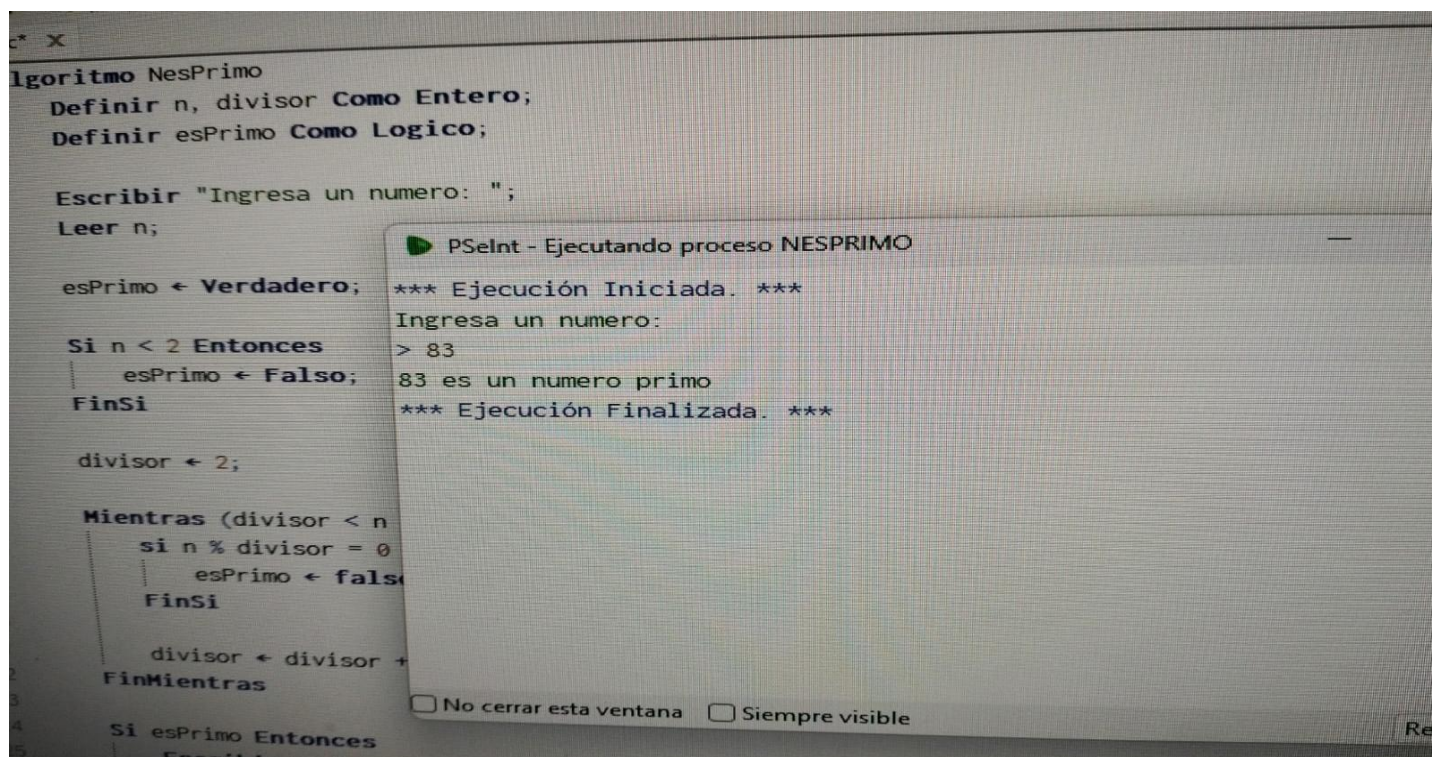
  divisor ← 2;
  Mientras (divisor < n)
    Si n % divisor = 0
      esPrimo ← Falso;
    FinSi
    divisor ← divisor + 1;
  FinMientras

  Si esPrimo Entonces
    Escribir n, " es un numero primo";
  FinSi

```

Output: Ingresar un numero: 8 no es un numero primo

12.- Ahora probamos con el número 83. Nos dice que sí es número primo.



```

Algoritmo NesPrimo
  Definir n, divisor Como Entero;
  Definir esPrimo Como Logico;

  Escribir "Ingresar un numero: ";
  Leer n;

  esPrimo ← Verdadero;
  Si n < 2 Entonces
    esPrimo ← Falso;
  FinSi

  divisor ← 2;
  Mientras (divisor < n)
    Si n % divisor = 0
      esPrimo ← Falso;
    FinSi
    divisor ← divisor + 1;
  FinMientras

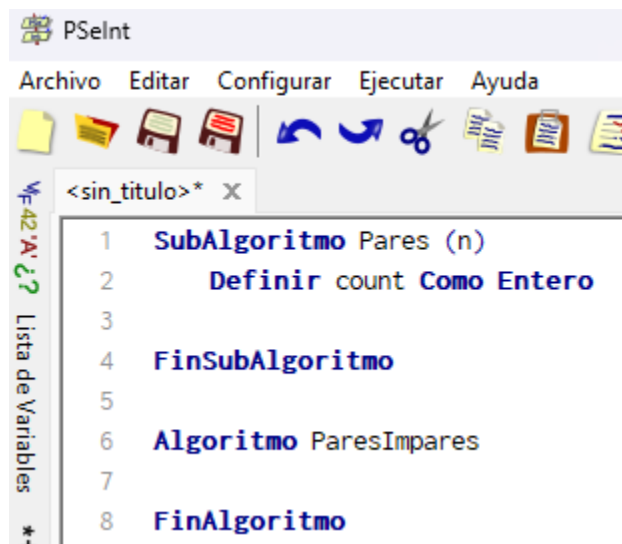
  Si esPrimo Entonces
    Escribir n, " es un numero primo";
  FinSi

```

Output: Ingresar un numero: 83 es un numero primo

## NÚMEROS PAR E IMPAR

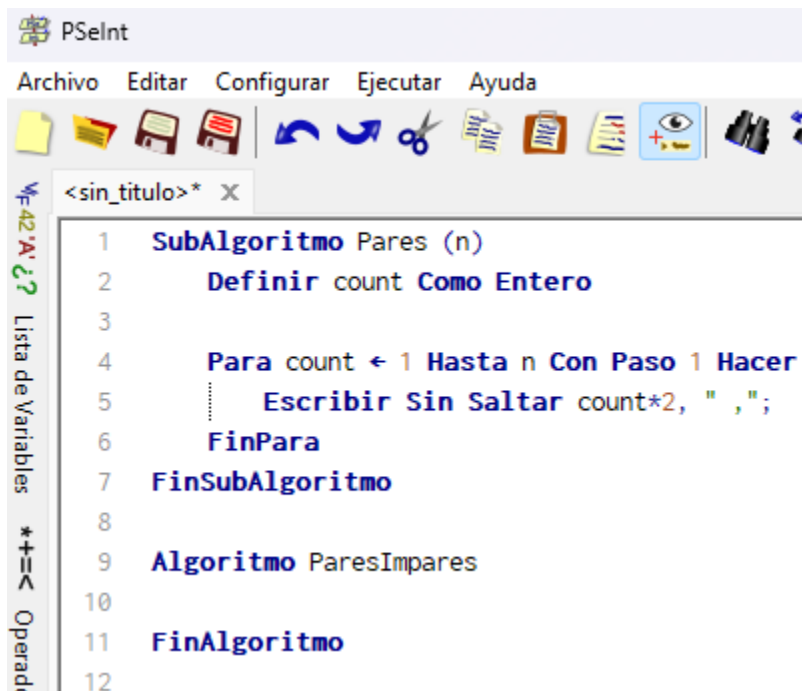
1.- Crearemos un nuevo "SubAlgoritmo" que se va a llamar Pares. No va a retornar ninguna variable, pero si va a recibir una variable n que va a ser la cantidad de números pares que va a mostrar, se va a "definir" una variable llamada count "como entero".



```

1  SubAlgoritmo Pares (n)
2      Definir count Como Entero
3
4  FinSubAlgoritmo
5
6  Algoritmo ParesImpares
7
8  FinAlgoritmo
  
```

2.- Se pondrá un "para" count y le asignamos 1 "hasta" el número n que queremos "con paso" 1 "hacer" "escribir". Añadiremos el "sin saltar" para que todo aparezca dentro de una misma línea. Se multiplica el valor de count por 2, se va concatenar.



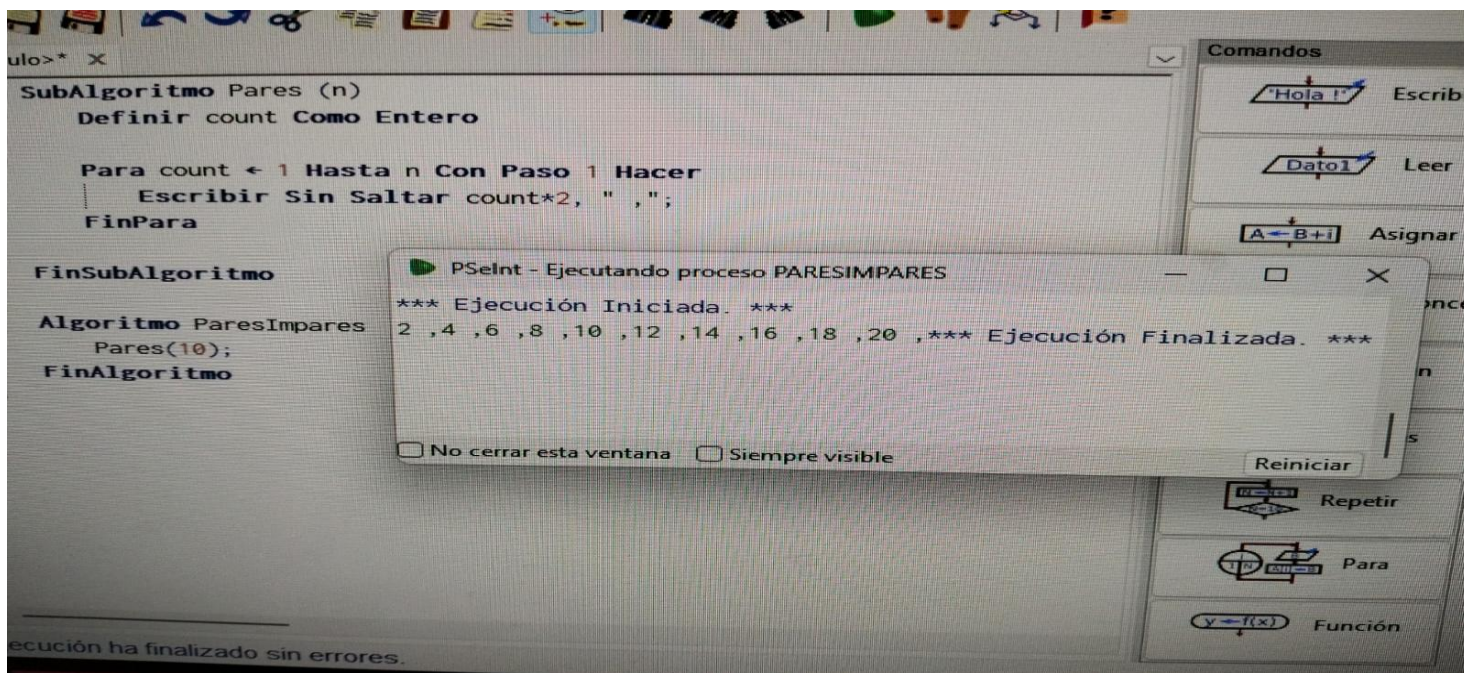
```

1  SubAlgoritmo Pares (n)
2      Definir count Como Entero
3
4      Para count ← 1 Hasta n Con Paso 1 Hacer
5          ..... Escribir Sin Saltar count*2, " ", ";
6      FinPara
7  FinSubAlgoritmo
8
9  Algoritmo ParesImpares
10
11  FinAlgoritmo
12
  
```



3.- Iremos ahora a nuestro algoritmo principal, el cual la vamos a llamar función par.

Vamos a decir que nos muestre 10 números pares. Nos tendría que mostrar 2,4,6 hasta el 20.



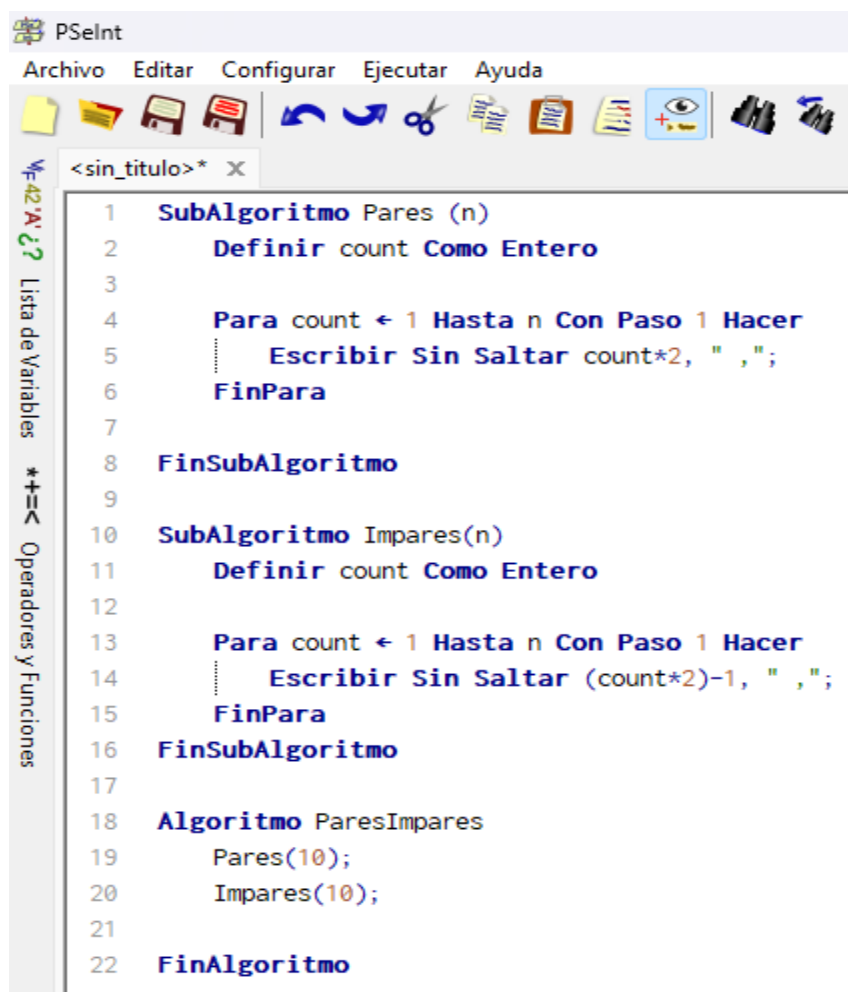
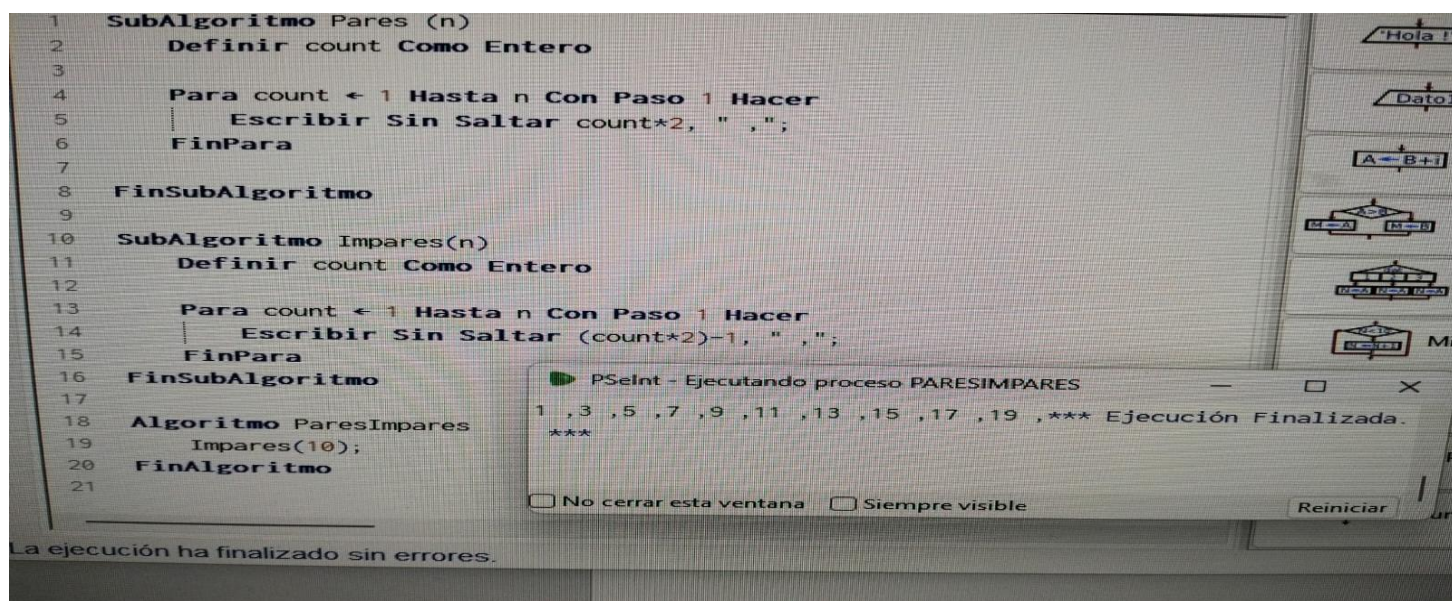
4.- Crearemos un nuevo “SubAlgoritmo” se va a llamar Impares. Igual no va a retornar ninguna variable, pero sí va a recibir una variable n que va a ser la cantidad de números pares que va a mostrar y para crear una secuencia de números impares, se pondrá la misma secuencia de números pares, solo que count multiplicado por 2 estará en paréntesis, menos 1.

```

8  FinSubAlgoritmo
9
10 SubAlgoritmo Impares(n)
11     Definir count Como Entero
12
13     Para count ← 1 Hasta n Con Paso 1 Hacer
14         ..... Escribir Sin Saltar (count*2)-1, " , ";
15     FinPara
16 FinSubAlgoritmo
17
18 Algoritmo ParesImpares
19     Pares(10);
20 FinAlgoritmo
21

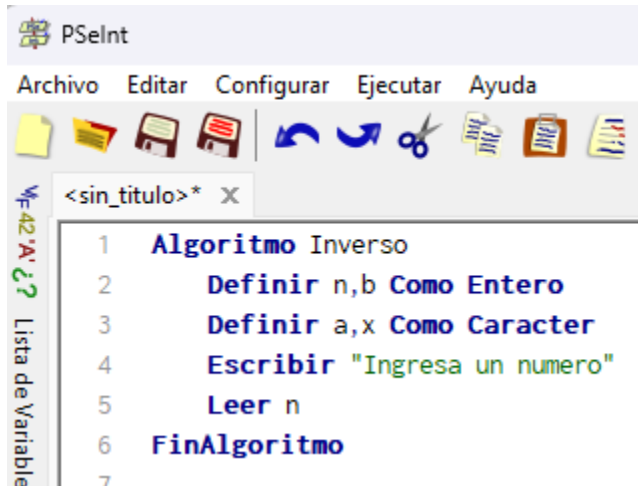
```

5.- Iremos ahora a nuestro algoritmo principal, el cual la vamos a llamar función Impares; vamos a decir que nos muestre 10 números Impares; nos tendría que mostrar 1,3,5 hasta el 19.



## NÚMEROS INVERTIDOS

1.- "Definir" la variable n y b, se va a declarar "como entero". Posteriormente se va a "definir" la variable a y x. Estas variables se declaran "como carácter" con la función de "escribir" "ingresa un número"; lo asignamos con la función de "leer" en la variable n.

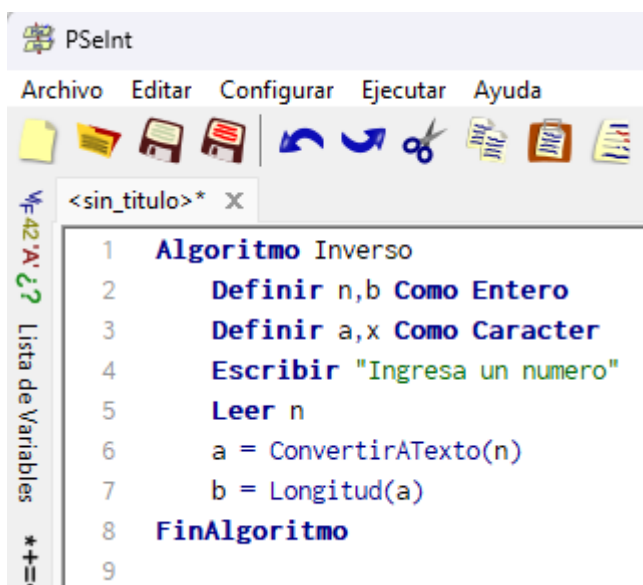


```

1  Algoritmo Inverso
2      Definir n,b Como Entero
3      Definir a,x Como Carácter
4      Escribir "Ingresa un número"
5      Leer n
6  FinAlgoritmo

```

2.- Una vez teniendo el número, lo vamos a convertir a texto; a es igual a "convertir a texto" y entre paréntesis colocamos la variable n. Posteriormente diremos que b es igual a "longitud" y entre paréntesis colocamos la variable a.



```

1  Algoritmo Inverso
2      Definir n,b Como Entero
3      Definir a,x Como Carácter
4      Escribir "Ingresa un número"
5      Leer n
6      a = ConvertirATexto(n)
7      b = Longitud(a)
8  FinAlgoritmo

```



3.- Comenzamos con la variable x es igual, con un espacio en blanco. Crearemos un ciclo “mientras” la variable b sea mayor a cero, “hacer” x es igual, a lo que tiene la variable x. Le vamos a concatenar un carácter de cadena o del número ingresado “subcadena” entre paréntesis a, b, b. Luego vamos a decir que b es igual a lo que tenga b menos 1.

```

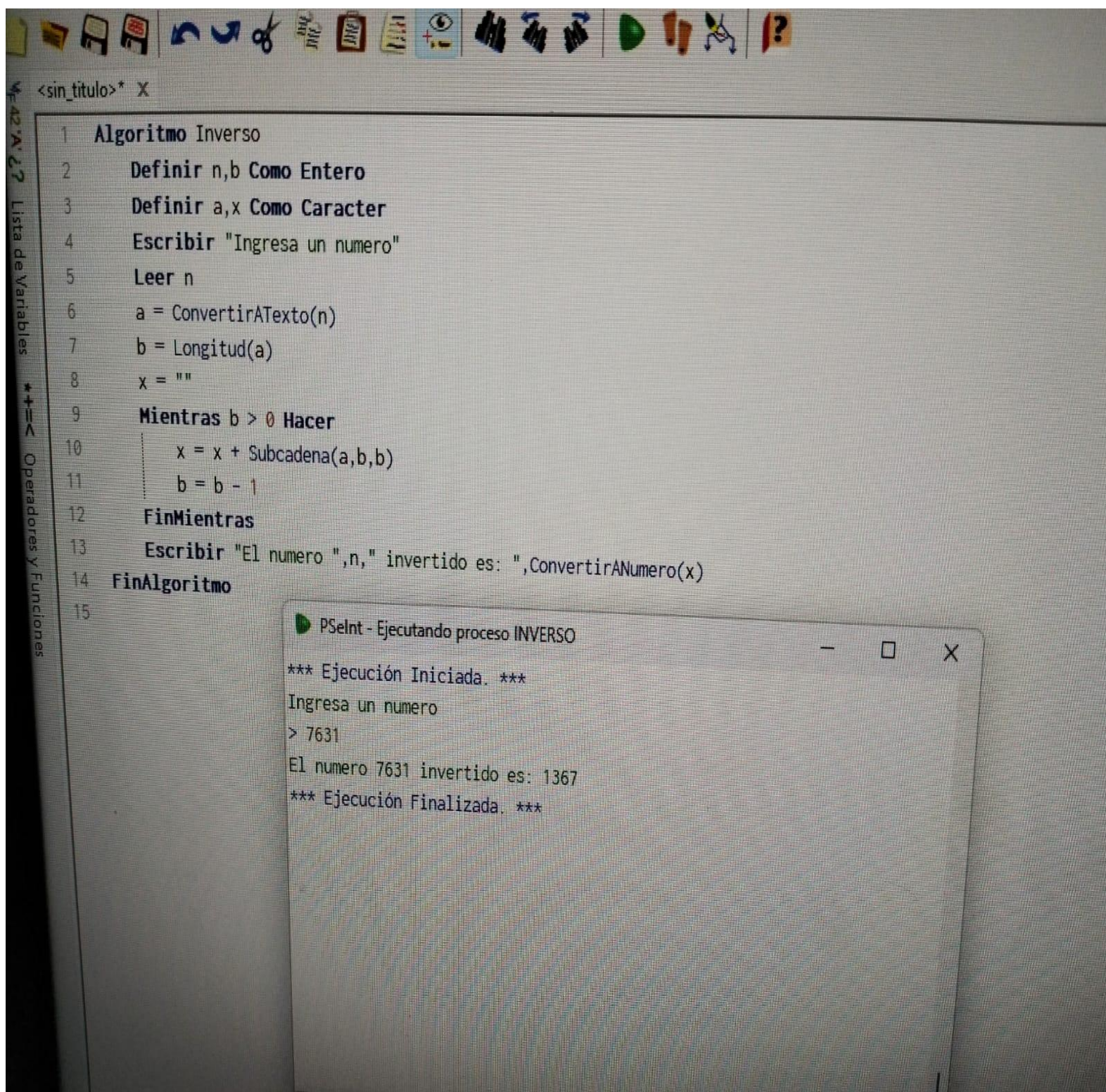
1  Algoritmo Inverso
2  Definir n,b Como Entero
3  Definir a,x Como Caracter
4  Escribir "Ingresa un numero"
5  Leer n
6  a = ConvertirATexto(n)
7  b = Longitud(a)
8  x = ""
9  Mientras b > 0 Hacer
10     x = x + Subcadena(a,b,b)
11     b = b - 1
12 FinMientras
13 FinAlgoritmo
  
```

4.- Fuera del ciclo, mientras ponemos la función “escribir”, vamos a decir “el número” que tenemos almacenado en la variable n “invertido es” “convertir a número” y entre paréntesis pondremos la variable x.

```

1  Algoritmo Inverso
2  Definir n,b Como Entero
3  Definir a,x Como Caracter
4  Escribir "Ingresa un numero"
5  Leer n
6  a = ConvertirATexto(n)
7  b = Longitud(a)
8  x = ""
9  Mientras b > 0 Hacer
10     x = x + Subcadena(a,b,b)
11     b = b - 1
12 FinMientras
13 Escribir "El numero ",n," invertido es: ",ConvertirANumero(x)
14 FinAlgoritmo
15
  
```

5.- Lo ejecutaremos de la siguiente manera.



The image shows a screenshot of a PSeInt IDE. The main window displays an algorithm for reversing a number. The algorithm is as follows:

```
1  Algoritmo Inverso
2  Definir n,b Como Entero
3  Definir a,x Como Caracter
4  Escribir "Ingresa un numero"
5  Leer n
6  a = ConvertirATexto(n)
7  b = Longitud(a)
8  x = ""
9  Mientras b > 0 Hacer
10     x = x + Subcadena(a,b,b)
11     b = b - 1
12 FinMientras
13 Escribir "El numero ",n," invertido es: ",ConvertirANumero(x)
14 FinAlgoritmo
15
```

Below the algorithm, a small window titled "PSeInt - Ejecutando proceso INVERSO" shows the execution output:

```
*** Ejecución Iniciada. ***
Ingresa un numero
> 7631
El numero 7631 invertido es: 1367
*** Ejecución Finalizada. ***
```



## CONCLUSIÓN

En general, los algoritmos son una herramienta muy importante que permite resolver problemas de forma sistemática y eficiente. Algunos aspectos a tener en cuenta sobre los algoritmos son:

- Son un conjunto de instrucciones ordenadas y finitas que se siguen para alcanzar un objetivo.
- Se utilizan en el desarrollo de software, en la inteligencia artificial y en la vida cotidiana.
- Son una pieza esencial de la inteligencia artificial, ya que permiten que la tecnología resuelva problemas de todo tipo, desde sumas hasta detectar cáncer.
- Son una herramienta que ayuda a desarrollar la inteligencia lógico-matemática.
- Son un paso inicial para adentrarse en el mundo del desarrollo de software y crear aplicaciones.
- Simulan los procesos de la inteligencia humana a través del aprendizaje, el razonamiento y la autocorrección.
- Cada paso de un algoritmo debe definirse de un modo preciso.
- Un algoritmo debe finalizar tras un número finito de pasos.

El uso de los algoritmos es importante en cualquier área debido a que no solo se utiliza en el desarrollo y programación de software, sino también en los problemas de la vida cotidiana, que ayuda a reforzar el razonamiento y análisis para comprender los conceptos y ejercicios.

## REFERENCIAS

Ramírez, A. (2022, 28 julio). *Algoritmo en informática: Qué es, cómo hacer uno y los diferentes tipos que existen*. Epitech Spain. <https://www.epitech-it.es/algoritmo-diferentes-tipos/>

Dani sable Programación. (2019c, octubre 9). *¡Aprende a programar desde cero con PseInt! / Interfaz de PseInt / Parte 3* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=MMnAwnOb4a8>

Programa Tutos. (2023, 24 agosto). *Creando un Detector de Números Primos en PSeInt* [Vídeo]. YouTube. [https://www.youtube.com/watch?v=lltV\\_mCoCXc](https://www.youtube.com/watch?v=lltV_mCoCXc)

Programa Tutos. (2023a, mayo 11). *Aprende a mostrar números pares e impares en PSeInt en 5 minutos* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=ZnDd6rXn8gY>

ALGORITMODETAREA. (2021, 16 agosto). *Algoritmo para invertir un numero en pseint* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=y0yd5Wiip7o>