

Actividad | #2 |

Diagrama de Flujo

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	7
NÚMERO PAR E IMPAR	9
NÚMEROS INVERTIDOS	12
CONCLUSIÓN	14
REFERENCIAS	15

INTRODUCCIÓN

En el presente documento se hablará acerca de los diagramas de flujo, que representa la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema.

Su correcta construcción es sumamente importante porque a partir del mismo se escribe un programa en algún Lenguaje de Programación. Los diagramas de flujo se construyen con símbolos y líneas que muestran la secuencia de pasos para realizar una tarea. Para crear un diagrama de flujo, se pueden seguir estos pasos:

1. Identificar los procesos, es decir, enumerar todas las etapas o pasos del proceso.
2. Seleccionar símbolos estándar para representar cada paso.
3. Ordenar los pasos en secuencia lógica, conectándolos con flechas.

Tomando en cuenta que un diagrama de flujo es una representación gráfica de los pasos que sigue un programa para procesar datos. En él, podemos utilizar varios patrones geométricos para ilustrar las numerosas acciones que lleva a cabo el programa. Un algoritmo es un procedimiento o conjunto de reglas que define cómo se debe ejecutar un programa.

DESCRIPCIÓN

Un diagrama de flujo es un diagrama que describe un proceso, sistema o algoritmo informático. Se usan ampliamente en numerosos campos para documentar, estudiar, planificar, mejorar y comunicar procesos que suelen ser complejos en diagramas claros y fáciles de comprender. En programación, se utilizan para diseñar el funcionamiento de programas, aplicaciones y sitios web.

Los diagramas de flujo se construyen con símbolos y líneas, y pueden variar desde simples y dibujados a mano hasta más complejos y creados por computadora. Algunos elementos que se utilizan en los diagramas de flujo son:

Inicio y final: Se indica con un rectángulo con esquinas redondeadas.

Asignación o definición: Se representa con un rectángulo y permite asignar un valor o definir una variable.

Datos de entrada o salida: Se representa con un recuadro con las esquinas inclinadas y una flecha hacia fuera.

Decisión: Se indica con un rombo y señala que puede haber más de un camino a seguir.

Es importante construir correctamente los diagramas de flujo, ya que a partir de ellos se escribe un programa en un lenguaje de programación.

JUSTIFICACIÓN

Los diagramas de flujo son una herramienta gráfica que representa los pasos, secuencias y decisiones de un algoritmo o programa, y son útiles para **justificar** su diseño y análisis:

- Permiten visualizar de forma clara y secuencial los pasos a seguir para desarrollar un algoritmo.
- Ayudan a explicar la lógica detrás de un programa antes de empezar a codificar.
- Son útiles para colaborar con otros en el desarrollo de un algoritmo.
- No requieren el conocimiento de un lenguaje de programación informática.

Los diagramas de flujo se pueden utilizar en diferentes disciplinas, como la programación, la economía, los procesos industriales, entre otras.

Algunos de los beneficios de utilizar diagramas de flujo son:

- Representar los flujos de trabajo actuales de manera comprensible.
- Supervisar los procesos o recoger datos.
- Identificar los potenciales de mejora.
- Crear un entendimiento común de los procesos.

Existen diferentes tipos de diagramas de flujo, como los de decisiones, lógicos, de trabajo, de producto, de proceso, de sistema y de casos de uso.

DESARROLLO

Para crear un diagrama de flujo en PSeInt, se puede seguir el siguiente procedimiento:

1. Dirigirse a la opción "dibujar diagrama de flujo".
2. Presionar el botón para generar el diagrama de flujo a partir del pseudocódigo escrito.

Para definir una variable en PSeInt, se debe:

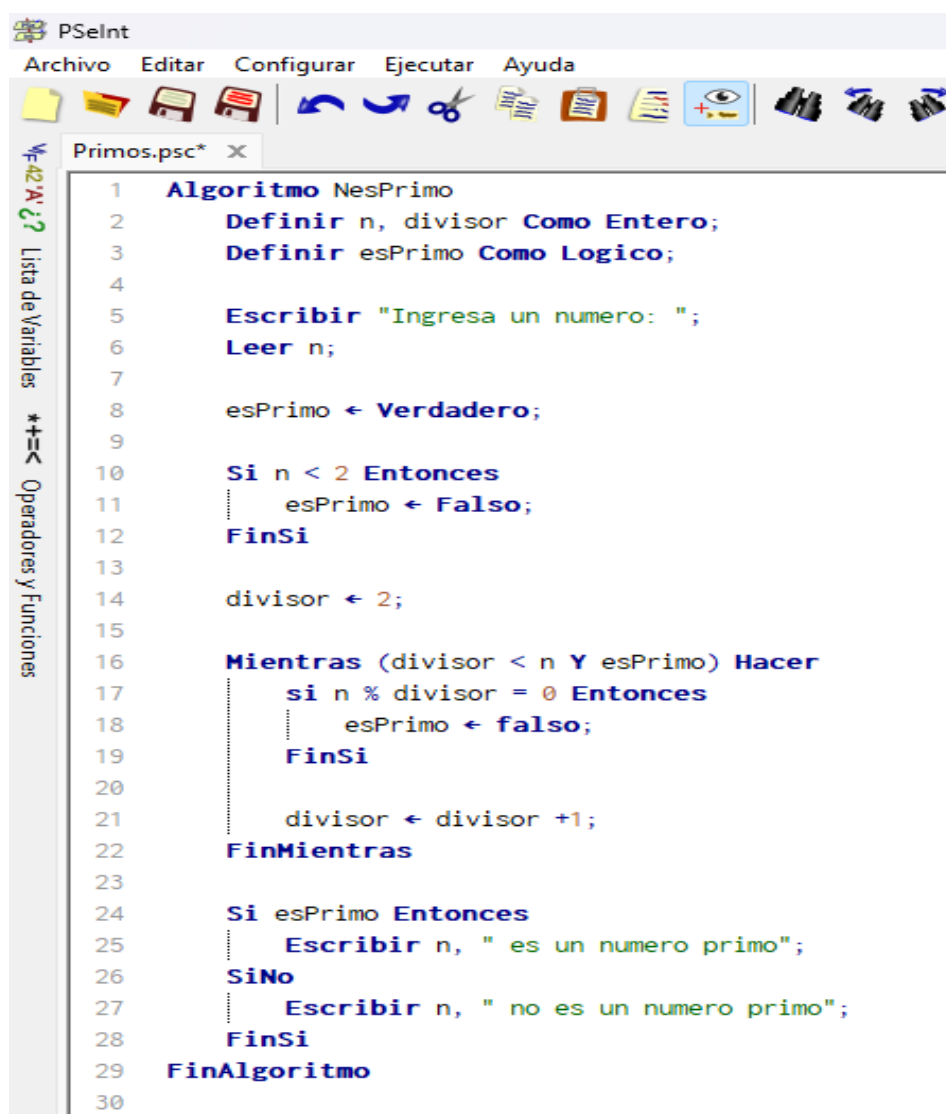
1. Colocar la palabra reservada "Definir".
2. Seguir con el nombre de la variable.
3. Añadir la palabra reservada "Como".
4. Indicar el tipo.

Para detener momentáneamente la ejecución de un algoritmo en PSeInt, se puede utilizar el botón "Pausar/Continuar".

NÚMEROS PRIMOS

Para determinar si un número es primo, se puede dividir el número entre 2, 3, 4, 5, y así hasta el número anterior a sí mismo. Si alguna de las divisiones es exacta, el número es compuesto, y si ninguna de las divisiones es exacta, el número es primo.

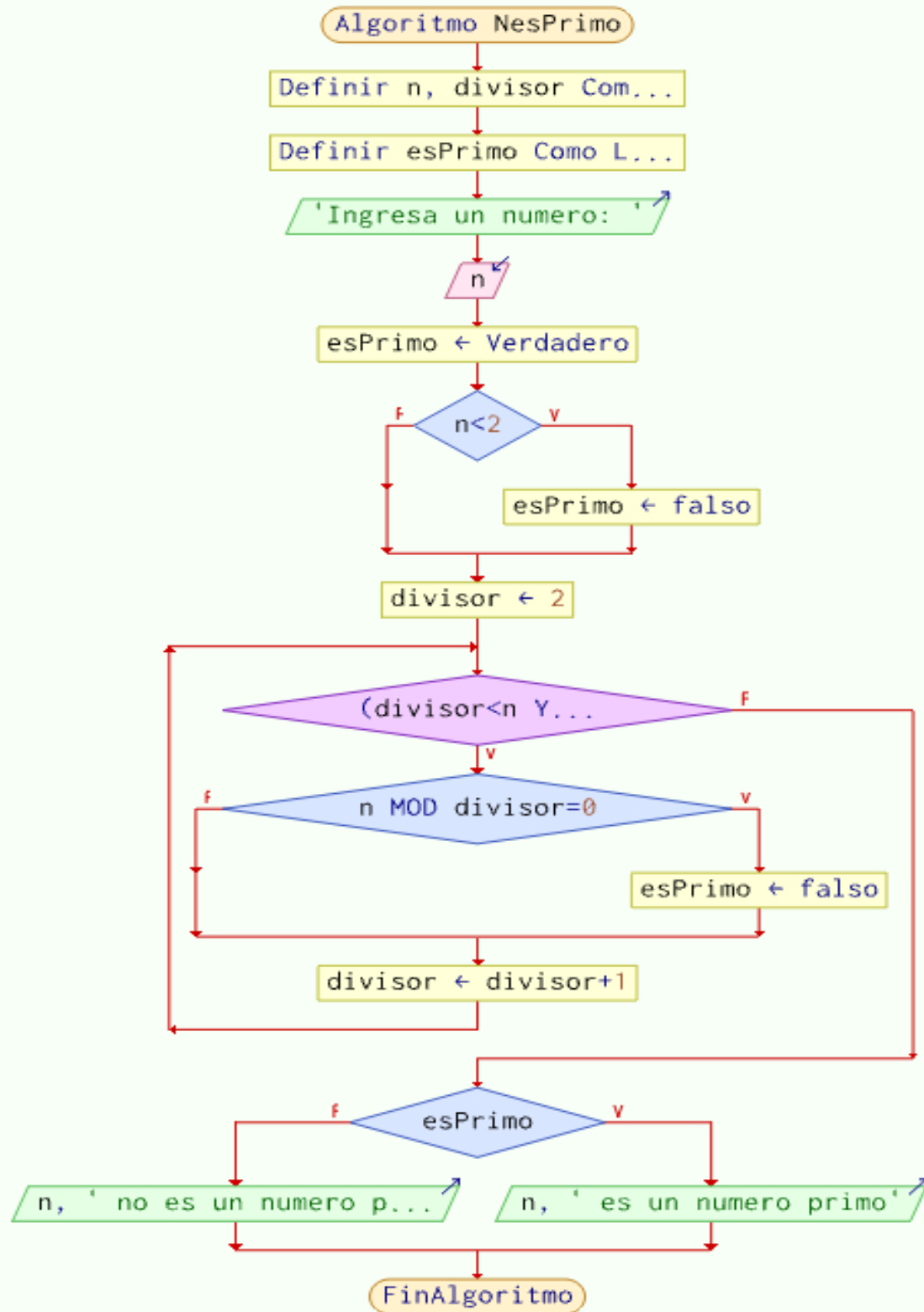
Un número primo es aquel que solo es divisible entre sí mismo y el 1. Si se intenta dividir un número primo por cualquier otro número, el resultado no es entero. A continuación, se verá el diagrama de flujo de la actividad 1 números primos.



The image shows a screenshot of the PSeInt (Pseudointerpreter) software interface. The window title is "PSeInt". 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 active file is "Primos.psc". The left sidebar shows a "Lista de Variables" (List of Variables) and "Operadores y Funciones" (Operators and Functions). The main editor displays the following pseudocode:

```

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         .....
12         esPrimo ← Falso;
13     FinSi
14
15     divisor ← 2;
16
17     Mientras (divisor < n Y esPrimo) Hacer
18         .....
19         si n % divisor = 0 Entonces
20             .....
21             esPrimo ← falso;
22         FinSi
23     FinMientras
24
25     Si esPrimo Entonces
26         .....
27         Escribir n, " es un numero primo";
28     SiNo
29         .....
30         Escribir n, " no es un numero primo";
31     FinSi
32 FinAlgoritmo
  
```

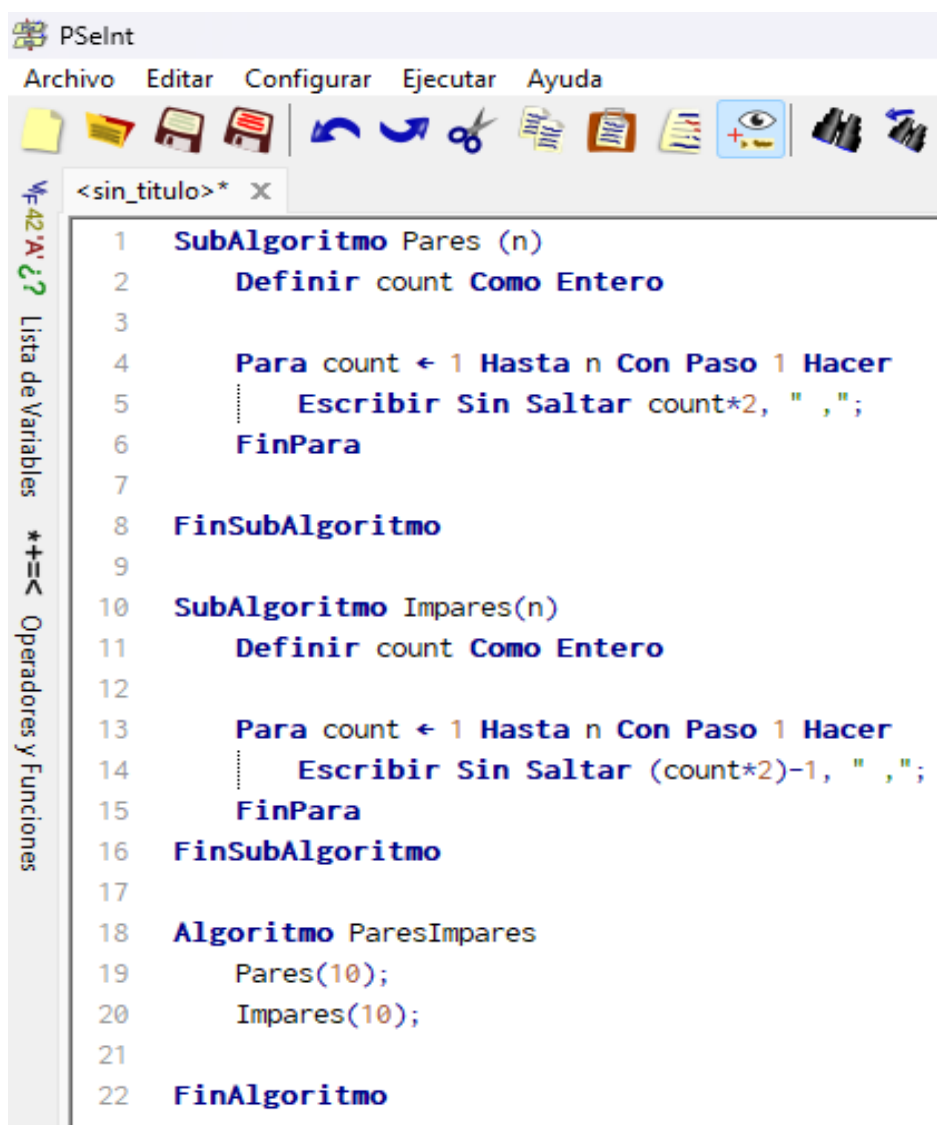


NÚMERO PAR E IMPAR

Un diagrama de flujo de números pares e impares es una representación gráfica de un algoritmo que muestra los pasos para identificar si un número es par o impar:

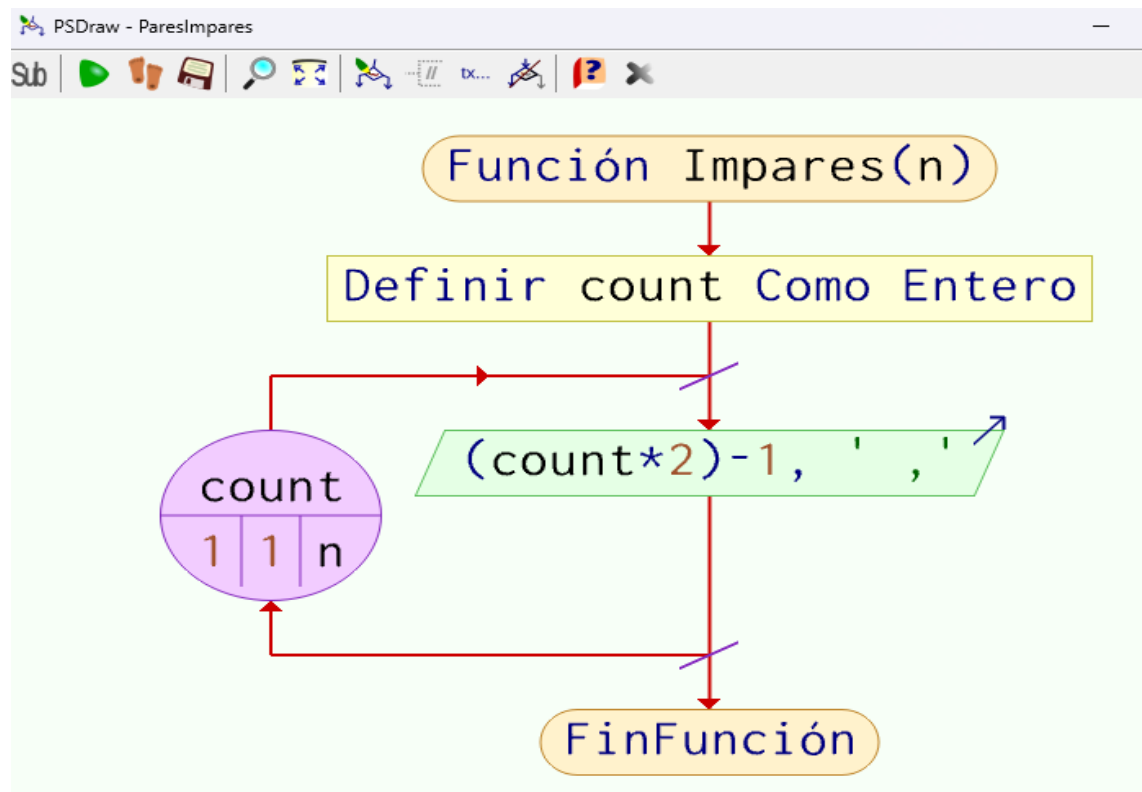
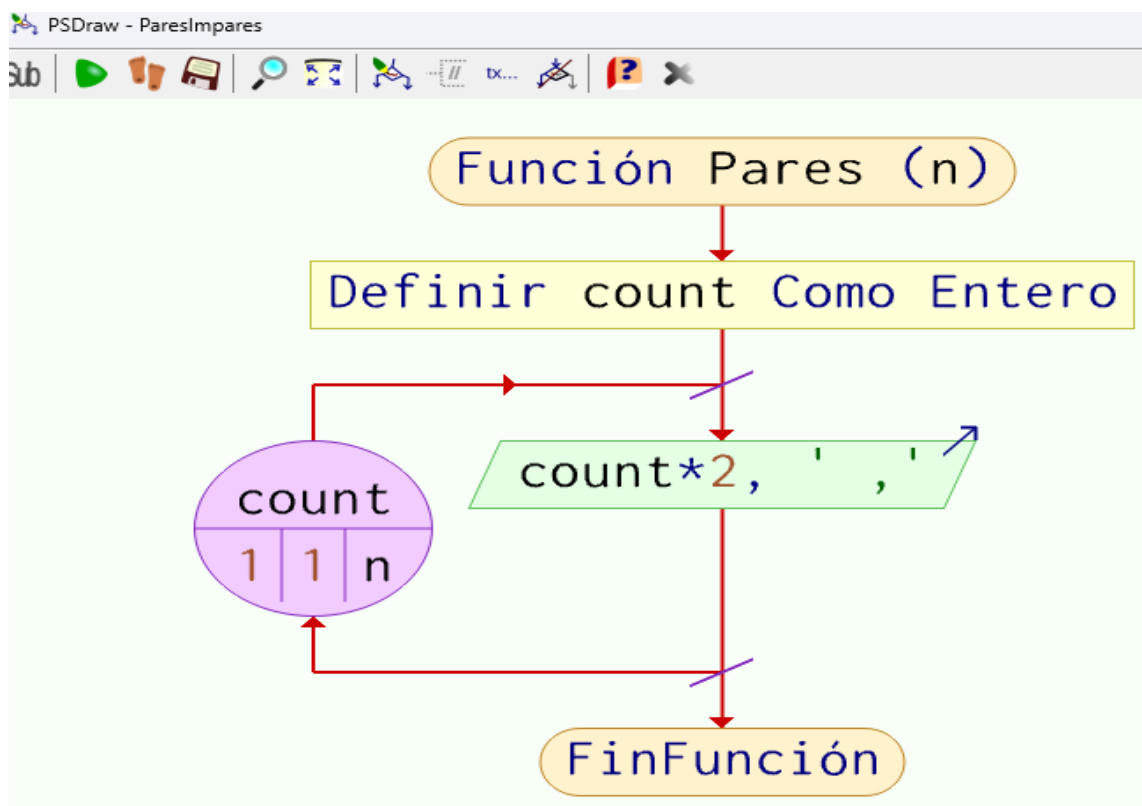
- Un número par es un entero que se puede dividir por dos sin dejar residuo.
- Un número impar es un entero que deja un residuo de uno al dividirlo por dos.

En la base de numeración 10 o 8, un número es par si su último dígito también es par. Por ejemplo, el número 6 es par porque su último dígito también es 6. A continuación, se verán los diagramas de flujos de la actividad 1 número par e impar.



```

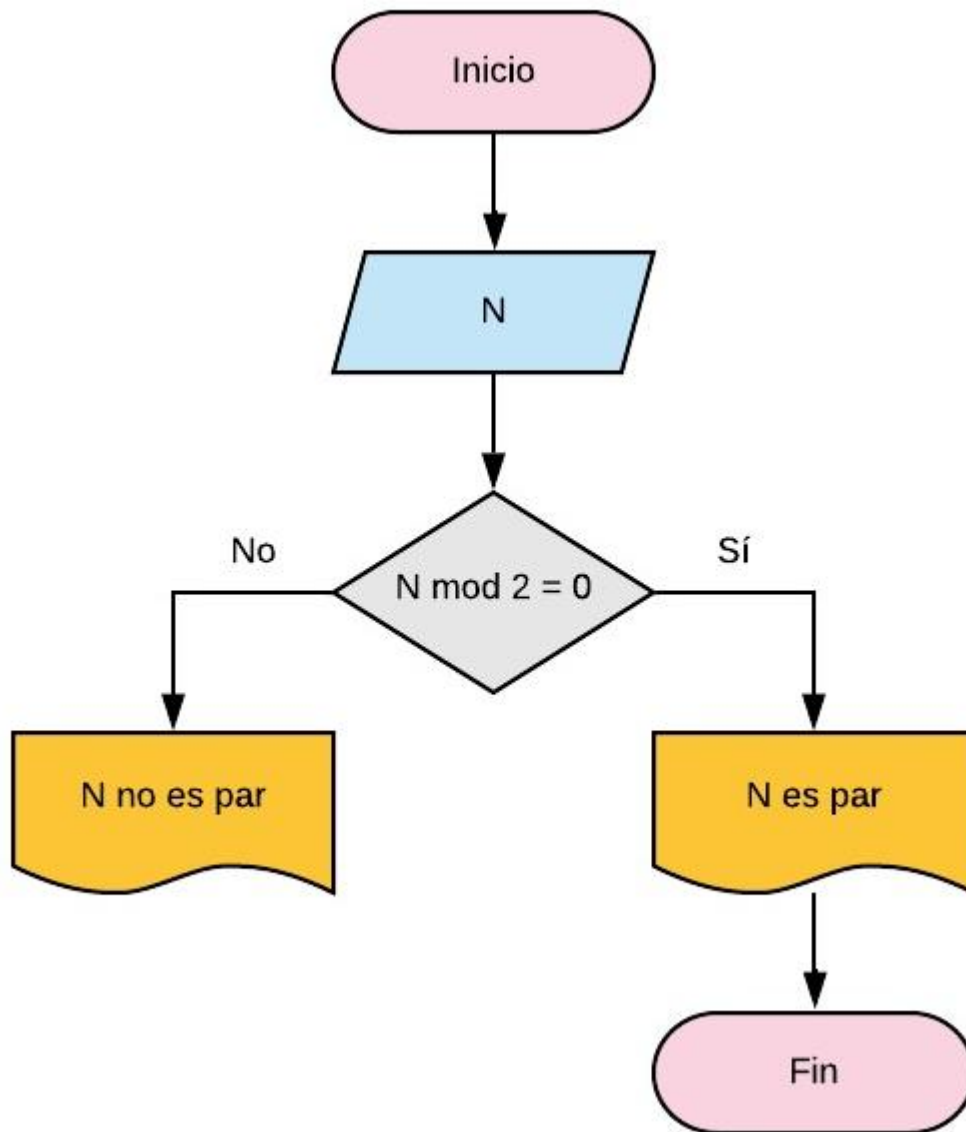
1  SubAlgoritmo Pares (n)
2      Definir count Como Entero
3
4      Para count ← 1 Hasta n Con Paso 1 Hacer
5          ..... Escribir Sin Saltar count*2, " ,";
6      FinPara
7
8  FinSubAlgoritmo
9
10 SubAlgoritmo Impares(n)
11     Definir count Como Entero
12
13     Para count ← 1 Hasta n Con Paso 1 Hacer
14         ..... Escribir Sin Saltar (count*2)-1, " ,";
15     FinPara
16 FinSubAlgoritmo
17
18 Algoritmo ParesImpares
19     Pares(10);
20     Impares(10);
21
22 FinAlgoritmo
  
```



Otra forma sencilla de saber si un número es par o impar en un diagrama de flujo es la siguiente:

1. Introducimos un número "N".

Si "N" se divide entre 2 y el residuo es 0 ($N \bmod 2 = 0$), entonces "N" es par. Caso contrario es impar.



NÚMEROS INVERTIDOS

Para invertir un número en PSeInt, se puede usar una fórmula que multiplica el número por 10 y le suma el dígito. Por ejemplo, si se tiene el número 2, se puede realizar la siguiente operación: $1 \text{ por } 10 = 10 + 2 = 12$.

En PSeInt, se puede definir una variable de la siguiente manera:

- Definir <NbreVariable> Como <tipo>
- El tipo puede ser Entero, Real, Lógico o Carácter

Algunos de los tipos de datos en PSeInt son:

- Real: Se utiliza para almacenar números con parte decimal.
- Carácter: Se utiliza para almacenar una secuencia de caracteres, como una palabra o una frase.
- Lógico: Se utiliza para almacenar valores booleanos, es decir, Verdadero o Falso.

Es importante tener en cuenta que este algoritmo está diseñado para números enteros, por lo que, si se desea invertir números decimales, se requerirá un enfoque diferente. Además, es recomendable realizar pruebas con diferentes números para asegurarse de que el algoritmo funcione correctamente en todos los casos. A continuación, se verá el diagrama de flujo de la actividad 1, números invertidos.

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

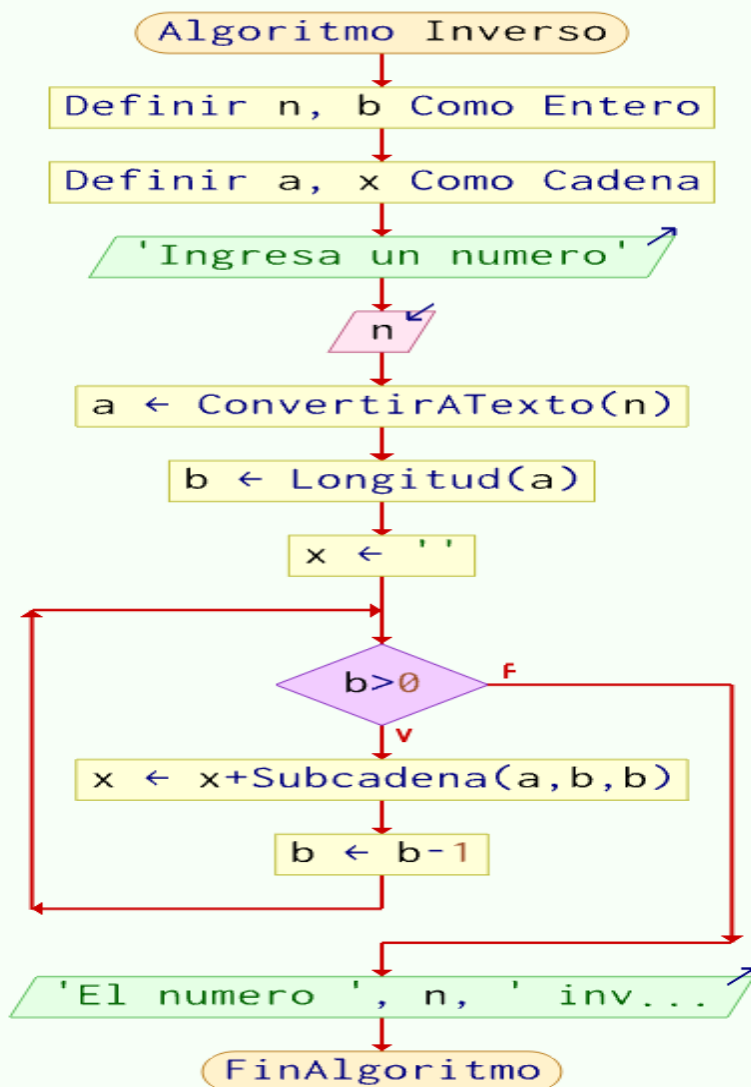
<sin_titulo>* X

```

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

```

Lista de Variables *+=< Operadores y Funcion



CONCLUSIÓN

En general, se podría decir que los diagramas de flujo son una herramienta valiosa para representar visualmente los algoritmos y procesos en la programación informática. Son útiles para:

- Comprender los procesos de un programa.
- Mejorar la comunicación entre los desarrolladores.
- Guiar el proceso de desarrollo de software.
- Explicar la lógica de un programa antes de codificar.
- Organizar la perspectiva general.
- Visualizar la ejecución del código.
- Mostrar la estructura de un sitio web o aplicación.
- Comprender cómo los usuarios navegan por un sitio web o programa.

Los diagramas de flujo son esquemas gráficos que muestran los pasos o procesos que se deben seguir para resolver un problema. Se construyen con símbolos como rectángulos, rombos, círculos, flechas, entre otros, que ayudan a visualizar la relación secuencial de las acciones.

La correcta construcción de los diagramas de flujo es muy importante, ya que a partir de ellos se escribe el programa en un lenguaje de programación.

REFERENCIAS

(S/f). Cibertareas.info. Recuperado el 1 de diciembre de 2024, de <https://cibertareas.info/pasos-para-elaborar-un-diagrama-de-flujo-en-pseint-programacion.html>

Algoritmo y Diagramas de flujo. (s. f.).

https://www.feriadetecnologia.com/arduino/algoritmo_y_diagramas_de_flujo.html

González, O. (2022, 21 enero). *¿Qué es un diagrama de flujo y para qué sirve? ¿Planificación y eficiencia en un solo paso!* <https://www.crehana.com>.

<https://www.crehana.com/blog/negocios/que-es-un-diagrama-de-flujo/>

Diagramas de flujo | Librería CATEDU. (s. f.).

<https://libros.catedu.es/books/fundamentos-de-programacion-estructurada-con-pseint-y-scratch/page/diagramas-de-flujo#:~:text=Un%20diagrama%20de%20flujo%20es,entrar%20directamente%20con%20el%20c%C3%B3digo.>