



Aprendizaje Autónomo (AA) Individual

Unidad 1

Tema:

Exploración y Uso de herramientas digitales para
pseudocódigo y diagramas de flujo

Docente:

Dr. Lopez Faican Lissette Geoconda

Carrera:

Computación

Estudiante Responsable:

Angulo Torres Yimmy Onner

Curso:

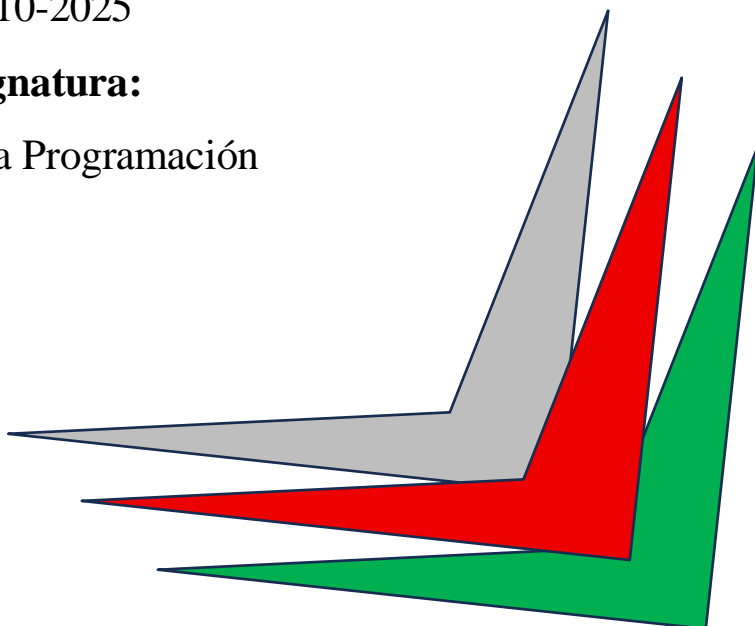
Primer Ciclo

Fecha:

12-10-2025

Asignatura:

Teoría de la Programación



1.- Investigar y explorar herramientas digitales para la construcción de algoritmos en pseudocódigo y diagramas de flujo:

Para poder llegar a la solución de un problema determinado mediante el uso de la computadora, podemos hacer uso de diferentes herramientas que nos sirven como intermediarios entre el usuario y el computador conocidos como compiladores.

Estos compiladores, son las diferentes herramientas que existen para diseñar código tanto en pseudocódigo, que son las sentencias o instrucciones que se le indica al computador para que realice mediante una serie de pasos tanto de manera secuencial, condicional o repetitiva.

O también mediante el diseño de diagramas de flujo, los cuales se caracterizan por utilizar formas geométricas para la representación de sus algoritmos, como lo podrían ser: elipses para representar el inicio y el final del algoritmo, siendo dos partes fundamentales en la programación, un cuadrado, representando los procesos que se pueden realizar dentro de él, además de flechas las cuales indican así donde esta direccionado nuestro algoritmo, y un sinnúmero de otras formas para la representación de algoritmos. [1]

Existen algunas herramientas muy populares para el uso de pseudocódigo y diagramas de flujo como lo serían PseInt, el cual se lo utilizan para inducir a las personas dentro del mundo de la programación debido a que es una herramienta un poco fácil para usar cuando se está comenzando a programar.

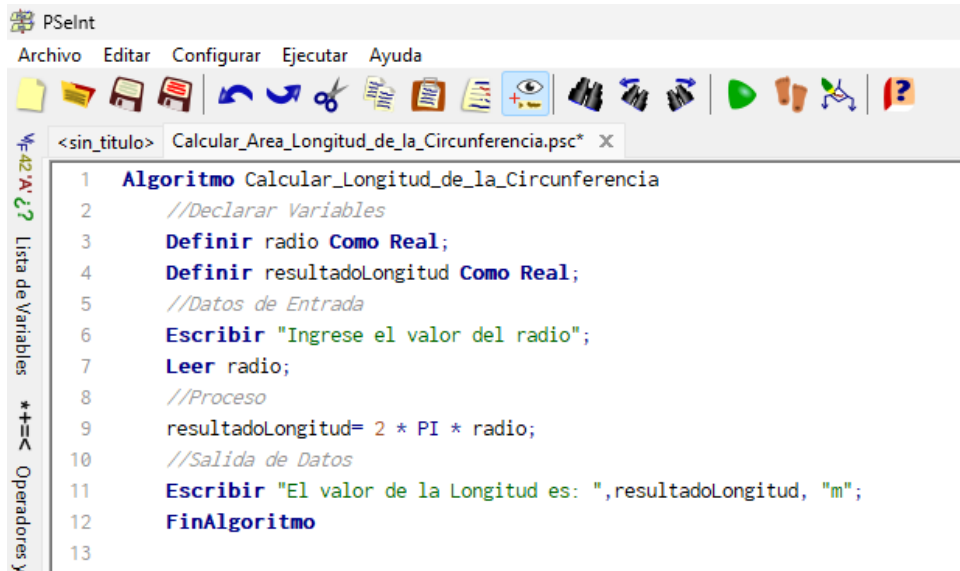
Otra herramienta conocida para el diseño de diagramas de flujo es FreeDFD, se trata de un editor e interprete creado exclusivamente para la implementación de diagramas de flujo, apto para Windows y Linux. Esta herramienta cuenta con varias instrucciones muy parecidas a las utilizadas en PseInt como lo serían los bucles, las estructuras secuenciales, repetitivas, entre otras. Además, todo algoritmo diseñado dentro de esta herramienta puede ser ejecutado dentro de la misma interfaz, facilitando la depuración de errores y la obtención de un código limpio, y aplicable al problema propuesto [2].

2. Seleccionar una herramienta que permita diseñar el algoritmo en pseudocódigo y diagrama de flujo o una herramienta para cada representación.

En esta ocasión, las herramientas que se utilizarán para el diseño de este algoritmo simple, serán las antes mencionadas que serían: PseInt y FreeDFD.

3.- Diseñar un algoritmo sencillo (de su elección), representándolo en pseudocódigo y diagrama de flujo en las herramientas seleccionadas.

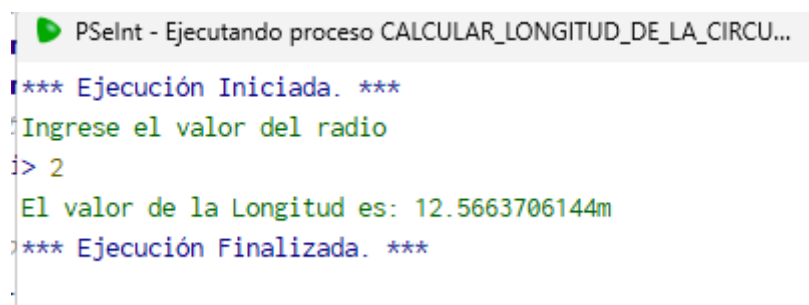
Algoritmo diseñado en PseInt:



```
1  Algoritmo Calcular_Longitud_de_la_Circunferencia
2      //Declarar Variables
3      Definir radio Como Real;
4      Definir resultadoLongitud Como Real;
5      //Datos de Entrada
6      Escribir "Ingrese el valor del radio";
7      Leer radio;
8      //Proceso
9      resultadoLongitud= 2 * PI * radio;
10     //Salida de Datos
11     Escribir "El valor de la Longitud es: ",resultadoLongitud, "m";
12     FinAlgoritmo
13
```

Para comprobar la utilidad de esta herramienta, se realizó un algoritmo para encontrar la longitud de una circunferencia en base a un radio dado, con una prueba de escritorio clave para cualquier algoritmo y esencial para la depuración de errores.

El mismo que necesita la presencia de 2 variables para funcionar como lo serian radio definida como Entero y de la variable resultadoLongitud definida como REAL y el valor de PI que es una constante en este caso. Se le pide al usuario que ingrese por teclado el valor de la variable para poder calcular la longitud, el mismo que se almacena en la variable radio, mediante la multiplicación de $2 \cdot \text{PI} \cdot \text{radio}$ se puede obtener el valor de la longitud el cual se almacena en la variable resultadoLongitud. Y al final este valor se presenta por pantalla mediante la instrucción: (Escribir "El valor de la Longitud es: ", resultadoLongitud, "m"); dando como resultado el valor de la longitud de la circunferencia.

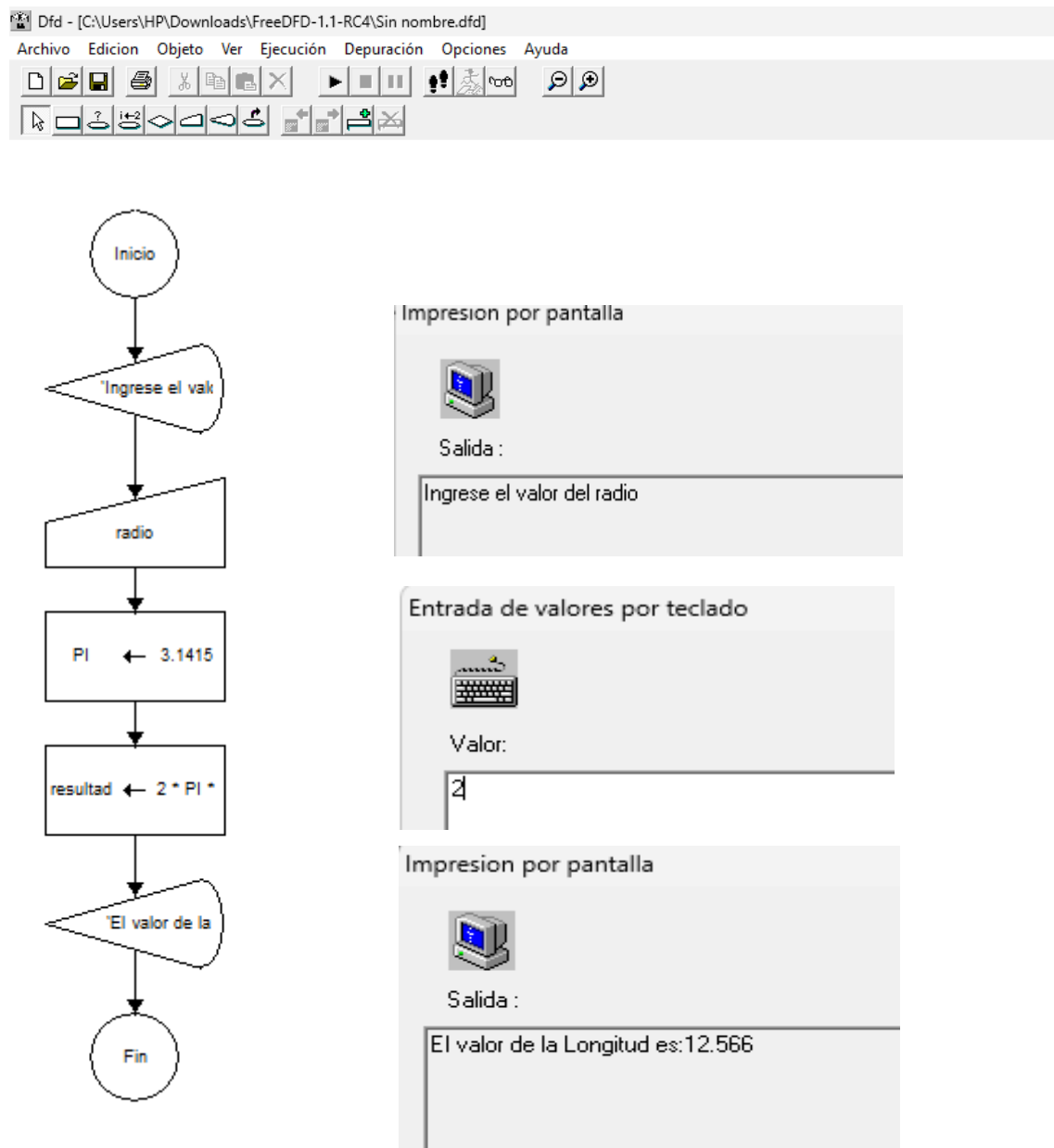


```
PseInt - Ejecutando proceso CALCULAR_LONGITUD_DE_LA_CIRCU...
*** Ejecución Iniciada. ***
Ingrese el valor del radio
> 2
El valor de la Longitud es: 12.5663706144m
*** Ejecución Finalizada. ***
```

Algoritmo diseñado en FreeDFD:

En esta ocasión el algoritmo aplicado es, hallar el valor de la longitud de una circunferencia mediante un radio dado.

El mismo que necesita la presencia de 2 variables para funcionar como lo serian radio, resultadoLongitud, PI, las cuales en esta ocasión no se las definió antes del proceso del algoritmo como si ocurrió con el algoritmo realizado en PseInt. Se le pide al usuario que ingrese por teclado el valor de la variable para poder calcular la longitud, el mismo que se almacena en la variable radio, mediante la multiplicación de $2 \cdot \text{PI} \cdot \text{radio}$ se puede obtener el valor de la longitud el cual se almacena en la variable resultadoLongitud. Y al final este valor se presenta por pantalla mediante la instrucción: ('El valor de la Longitud es:', resultadoLongitud) dando como resultado el valor de la longitud de la circunferencia y un dato extra en FreeDFD al momento de presentar algún texto por pantalla, este mismo se debe escribir dentro de comillas simples (''), debido a que esta herramienta no admite otro símbolo dentro de esta sentencia. [3]



4.- Elaborar un informe que sintetice los resultados:

Para el algoritmo realizado en PseInt:

PseInt, cuenta con una gran variedad de características las cuales nos permiten diseñar, elaborar y ejecutar una gran cantidad de algoritmos de una manera rápida y eficiente las cuales son:

- ✱ Gran versatilidad al momento de escribir las sentencias para la elaboración del algoritmo
- ✱ Esta herramienta nos indica cuales son palabras reservadas del lenguaje de programación, esto nos ayuda a conocer algunas sentencias que nos serán de utilidad en el futuro, como también, a prevenir el uso de estas palabras como una variable o constante dentro de nuestro algoritmo.
- ✱ Otra ventaja de PseInt, es su fácil sistema de depuración de errores, que me quiero decir con esto, me refiero a la forma para encontrar y depurar los errores dentro del programa, debido a su interfaz de alerta, señalando en que línea de código o en que parte de nuestro algoritmo existe algún error.
- ✱ Además, nos ofrece una salida por pantalla por dinámica y fácil de comprender para las personas que recién están entrando en el mundo de la programación, siendo una base muy buena para otros lenguajes de alto nivel como lo serian Java NetBeans, C++, Python, entre otros.

Para el algoritmo realizado en FreeDFD:

FreeDFD es una herramienta muy buena, para el diseño, elaboración y ejecución de diagramas de flujo, el cual cuenta con una interfaz de nivel medio, que quiero decir con esto, que es un poco complicada para entender, pero que, con un poco de investigación, se podrán realizar algoritmos dentro de esta herramienta sin muchas complicaciones.

- ✱ Ideal para aprender lógica de programación, estructuras de control y diseño algorítmico visual. Facilita la comprensión de cómo se organiza el flujo de un programa antes de escribir código en un lenguaje real. Además, ayuda a los estudiantes a desarrollar el pensamiento lógico y a resolver problemas de manera estructurada.
- ✱ Un punto negativo, es que al momento de querer introducir un nuevo cuadro o elipse, o alguna forma para seguir con el algoritmo, tenemos que ponerla encima de un cuadrado azul señalado en la misma dentro del algoritmo.

📖 Declaración del Uso de la IA en la tarea:

- Se utilizó la IA para poder verificar la sintaxis para la elaboración de las diferentes referencias bibliográficas en este documento, además de proporcionarme una mejor descripción de lo que es FreeDFD, la cual no se añadió en este trabajo, pero me sirvió, como una retroalimentación a los conocimientos que ya contaba antes de realizar este taller.

📖 Conclusiones:

- El desarrollo de algoritmos mediante **pseudocódigo y diagramas de flujo** representa un pilar esencial dentro de la creación de soluciones informáticas, ya que permite organizar de forma lógica y estructurada el pensamiento del programador antes de traducirlo a un lenguaje de programación. Gracias a estas herramientas, se logra una mejor comprensión del problema, se minimizan los errores y se mejora la eficiencia del proceso de desarrollo.
- El empleo del **pseudocódigo** favorece la precisión y el orden en la elaboración de programas, al describir de manera secuencial las acciones que debe ejecutar el sistema. Su sintaxis simple y cercana al lenguaje natural facilita el aprendizaje y la adaptación hacia lenguajes de programación más complejos, como Python, C++ o Java.
- Por su parte, los **diagramas de flujo** aportan una perspectiva visual clara del funcionamiento del algoritmo, mostrando la secuencia de operaciones, decisiones y bucles dentro del proceso. Esta representación gráfica ayuda a detectar fallos lógicos, mejora la comunicación entre los desarrolladores y contribuye a una mejor documentación técnica del software.

Referencias Bibliograficas:

[1] Vanessa Valverde, Fundamentos de Programacion con DFD.pdf, Escuela Superior Politecnica Chimborazo, octubre, 2023
https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=http://cimogsys.espoloch.edu.ec/direccion-publicaciones/public/docs/books/2024-01-23-142632-fundamentos_de_programaci%C3%B3n_con_DFD.pdf&ved=2ahUKEwjj-N-I15-QAxX9SjABHUrNM3Y4ChAWegQIFhAB&usg=AOvVaw2R-jGNZHQGY1CfNtHs6eCM

[2] Jocelyn, "FREEDFD," Infogram, [En línea]. Disponible en: <https://infogram.com/freedfd-1h7z2l80zwewg6o>

[3] Rai Carbajal, Diagrama de flujo en Dfd, Youtube, 23 de Marzo del 2020, [En línea], Disponible en: https://www.youtube.com/watch?v=_rLVZB4YPw8