ENES
JURIQUILLA

Universidad Nacional Autónoma de México

Escuela Nacional de Estudios Superiores, Campus Juriquilla

Práctica 3: Solución de problemas y Algoritmos

Integrantes del equipo:

- *Ricardo Antonio Posada Herrera 425127216*
- *Stefan Fackelman Robles 425007938*
- *Gael Israel Sánchez Arámburo 425034572*

Profesor: Ingeniero Andrés David Flores Ferro

Segundo Semestre

Para viernes 21 de Febrero de 2025

Actividad en clase 0 de 25 pts	
Protocolo de práctica	**60 de 75 pts**
Portada	10 de 10 pts
Introducción	13 de 15 pts
Desarrollo	19 de 25 pts
Conclusiones	10 de 15 pts
Referencias	08 de 10 pts
CALIFICACIÓN TOTAL	**Entrega 60 pts**

Fecha de revisión:

01 / 03 / 205

Observaciones:

- Introducción se agrega contenido nuevo pero muy poco de la clase y no termina de conectar bien.
- En el desarrollo solo se menciona el contenido pero no hay desarrollo de teoría de lo que se hizo o porque, especificación indica descripción paso a paso de lo realizado y de su porque.
- Agregar dato de tipo de archivo consultado en referencias y la fecha de consulta tambien es relevante [Web,PDF, Libro, etc]

Calificación:

60 de 100 pts

INTRODUCCIÓN

En el mundo actual, la informática se ha convertido en una herramienta indispensable en prácticamente todos los ámbitos de la vida. Desde la gestión de grandes empresas hasta la comunicación personal, el software juega un papel crucial en la resolución de problemas y la optimización de procesos. Sin embargo, el desarrollo de software de calidad no es una tarea trivial. Requiere de un enfoque sistemático y disciplinado que permita garantizar la creación de productos fiables, eficientes y que satisfagan las necesidades de los usuarios.

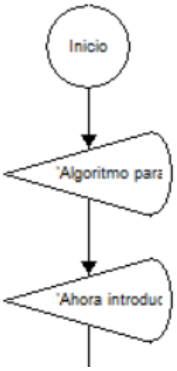
Un problema informático puede ser definido como un conjunto de instancias que se corresponden con un conjunto de soluciones, junto con una relación que asocia a cada instancia del problema un subconjunto de soluciones.

La Ingeniería de Software, según el IEEE (Institute of Electrical and Electronics Engineers), se define como "la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software". Esta disciplina provee los métodos y técnicas necesarias para llevar a cabo cada una de las etapas del ciclo de vida del software, desde la planificación y estimación del proyecto, hasta el análisis de requerimientos, diseño, codificación, pruebas y mantenimiento.

El uso de principios sólidos de ingeniería es esencial para obtener un software que sea económicamente viable y que funcione de manera eficiente. En este contexto, resulta fundamental comprender los fundamentos de la Ingeniería de Software y su aplicación en la resolución de problemas informáticos. A lo largo de este [documento/presentación/análisis], exploramos en detalle cada una de las etapas del ciclo de vida del software, así como los principios y técnicas que permiten garantizar la creación de productos de calidad.

DESARROLLO

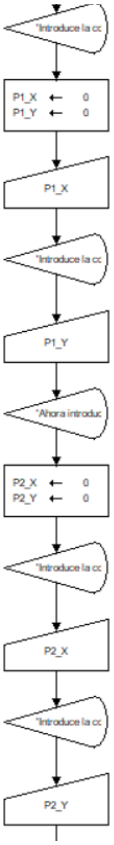
DFD 2 - Algoritmo de distancia entre 2 puntos



Iniciamos nuestro algoritmo escribiendo 2 mensajes para el usuario:

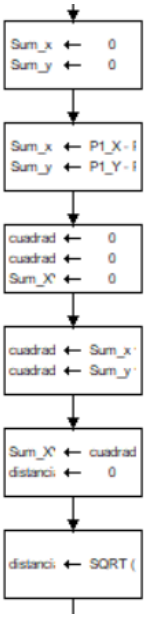
1. Uno que declare el propósito del algoritmo con un mensaje ‘Algoritmo para calcular la distancia entre 2 puntos’

2.Otro mensaje que indique que ahora se le pedirá al usuario que introduzca las coordenadas de los 2 puntos cuya distancia desea calcular.



Procedemos a declarar 4 variables correspondientes a las coordenadas de cada punto: P1_X, P1_Y, P2_X y P2_Y.

Mediante un mensaje antes de cada lectura, le indicamos al usuario que coordenada debe de introducir.

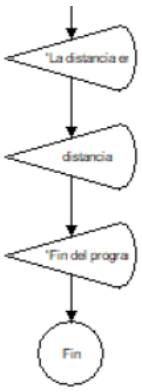


Ahora procedemos a las operaciones correspondientes.

Primero declaramos 2 nuevas variables que contendrán la suma de las coordenadas en X y en Y de los 2 puntos.

Se calcula la diferencia entre las coordenadas X y Y de los puntos, juegos se elevan al cuadrado las diferencias calculadas, después se suman los cuadrados de las diferencias y, finalmente, se calcula la raíz cuadrada de la suma anterior para obtener la distancia euclidiana. Los resultados de cada una de estás operaciones se van guardando en nuevas variables ya declaradas.

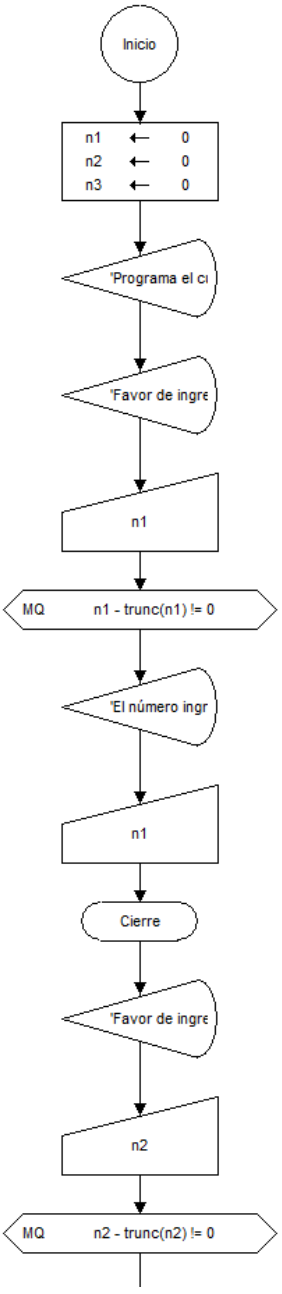
Por último, guardamos el resultado final en una variable ya declarada como distancia.

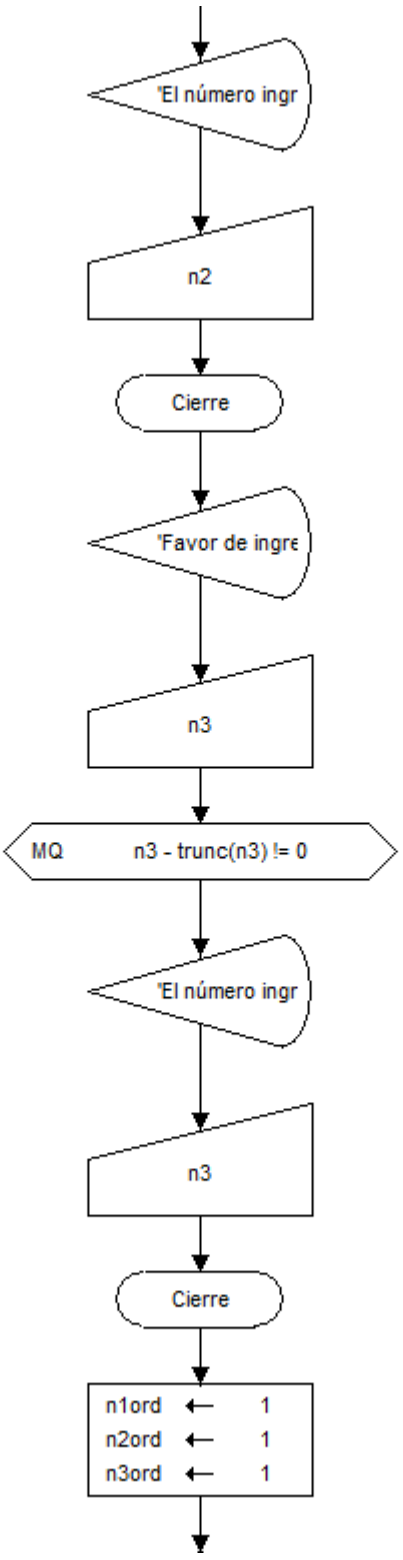


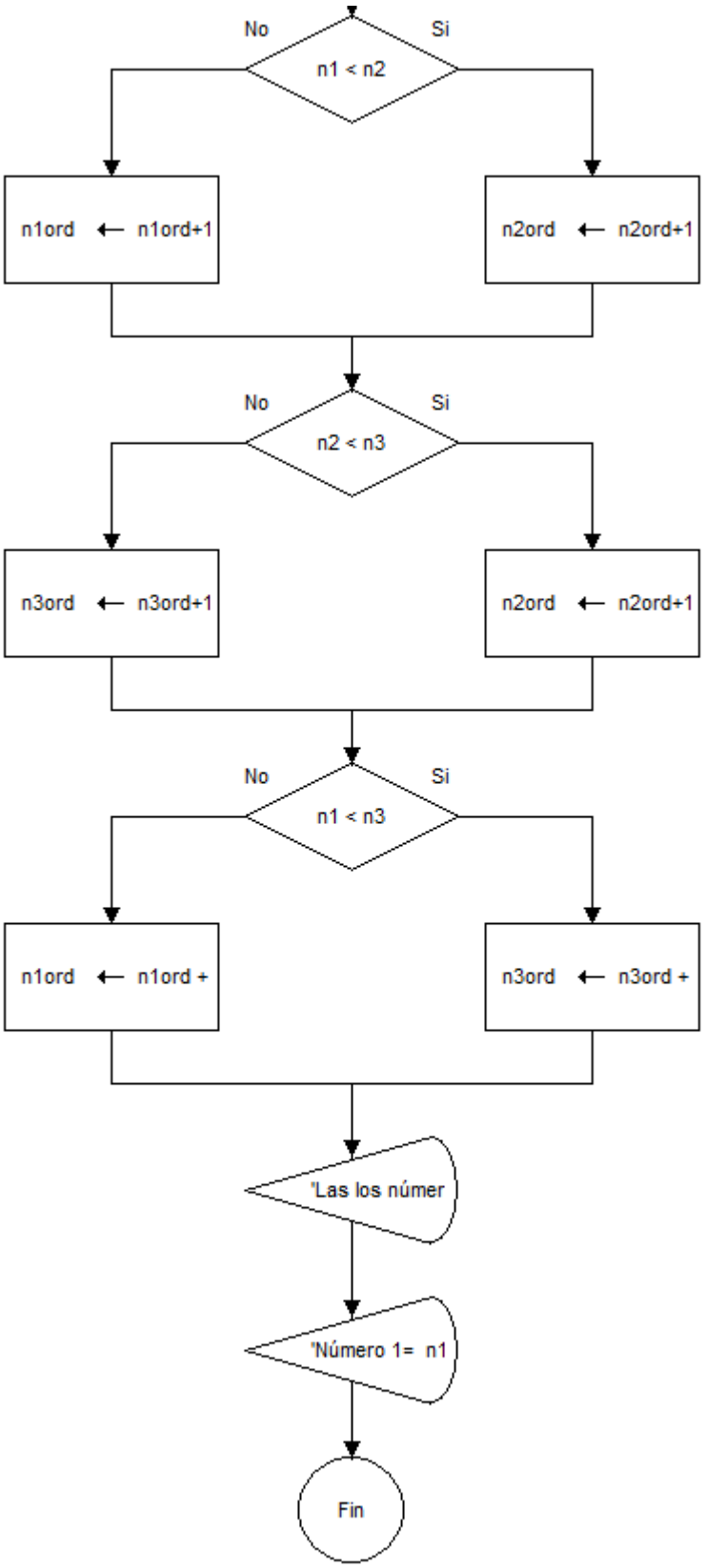
Para terminar nuestro algoritmo, escribimos un mensaje al usuario indicando al usuario que la distancia entre sus los 2 puntos indicados es el valor guardado en la variable “distancia”.

Finalmente, indicamos que el programa ha llegado a su fin.

El programa para ordenar 3 valores enteros se ve de la siguiente manera:




'El número ingr
n2
Cierre
'Favor de ingre
n3
MQ
n3 - trunc(n3) != 0
'El número ingr
n3
Cierre
n1ord ← 1
n2ord ← 1
n3ord ← 1



CONCLUSIONES

Ricardo:

Construir algoritmos requiere comprender el funcionamiento de diversas operaciones básicas de programación, que van más allá de asignar entradas, salidas u operar con variables. También implica el uso de condicionales y ciclos que permiten desarrollar algoritmos más complejos. Al emplear diagramas de flujo durante esta plática, hemos logrado visualizar con mayor claridad la progresión lógica del algoritmo y cómo cada función se interrelaciona, facilitando así su comprensión y diseño.

Stefan:Esta práctica nos ha dejado entender las estructuras de flujo y el pensamiento algorítmico más a fondo, resaltando las dificultades que pueden existir de limitaciones en las herramientas disponibles.

Gael: En esta práctica pude entender la complejidad de algunos elementos algorítmicos y el porqué es bueno saber el orden y claridad de las cosas que solicitas para así tener una respuesta eficaz de lo solicitado

REFERENCIAS:

Integration of Wind Power and Wave Power Generation Systems Using a DC Microgrid. (2015, 1 agosto). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/6948259