|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor(a):* | Ariel Adara Mercado Martínez |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 2 |
| *No. de práctica(s):* | 4 |
| *Integrante(s):* | 1 |
| *No. de lista o brigada:* | 29 |
| *Semestre:* | Segundo |
| *Fecha de entrega:* | 12 de marzo de 2025 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Indice**

Introducción ….. 3

Imágenes ….. 4

Preguntas ….. 5

Conclusión ….. 7

Introducción

El desarrollo de diagramas de flujo es una herramienta fundamental para la comprensión y la representación de los procesos algorítmicos,a través de estos diagramas podemos visualizar de manera clara y ordenada las secuencias de acciones necesarias para resolver un problema o realizar un proceso específico.Los diagramas de flujo permiten representar cada paso de un algoritmo de forma gráfica, facilitando la comprensión de su estructura lógica.El objetivo de esta practica fue el diseño de diagramas de flujo que describan las soluciones algorítmicas de problemas dados por la profesora, promoviendo nuestra capacidad para analizar y resolver problemas mediante la programación y el pensamiento lógico.

El manejo de variables acumulativas también es un aspecto central de esta actividad,las variables acumulativas son aquellas que permiten almacenar resultados progresivos de operaciones (la suma acumulada de varios números), lo cual es fundamental cuando se trabajan con secuencias de datos.

Los bucles while y for son herramientas para realizar repeticiones de un conjunto de acciones bajo ciertas condiciones,estos permiten automatizar tareas repetitivas y trabajar de manera más eficiente,ahorrando tiempo al evitar la repetición manual de pasos.

Las estructuras condicionales como if y else,permiten tomar decisiones dentro de un algoritmo bajo ciertas condiciones,ejecutando diferentes bloques de código,esta capacidad de tomar decisiones se usa para abordar una amplia variedad de problemas en la programación.

Desarrollo

Como primer actividad se requeria realizar un diagrama de flujo para determinar si un numero es primo por medio del uso de bucles y condiciones estructurales.

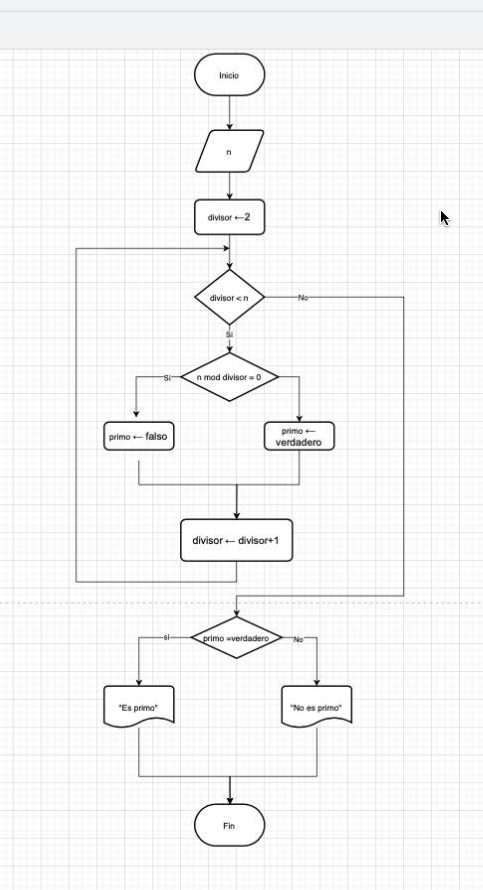


Ilustración 1

Para despues continuar con la serie Fibonacci,generando la serie hasta un limite proporcionado por el usuario por medio de bucles otra vez y manejo de variables acumulativas.

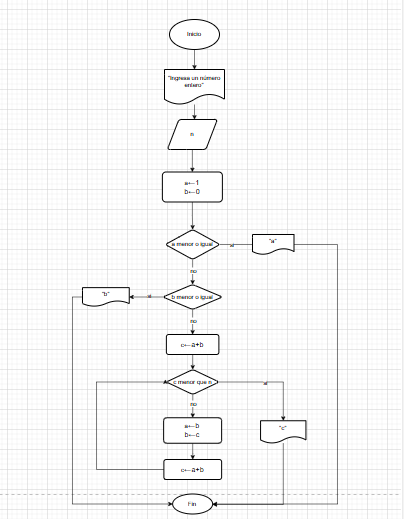


Ilustración 2

Y finalmente con un diagrama de flujo que calcule el factorial de un numero ingresado por el usuario.

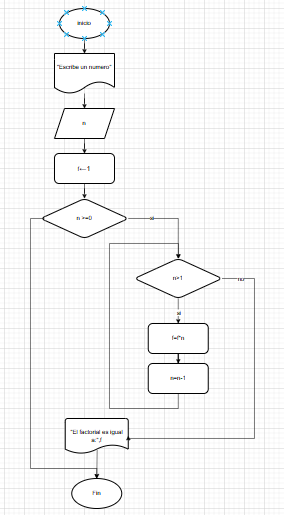


Ilustración 3

Preguntas para reflexionar

Sobre la estructura y lógica del diagrama de flujo:

– ¿Que elementos básicos de un diagrama de flujo utilizaste en los ejercicios?

Inicio,fin,leer,imprimir,asignaciones,condicionales,proceso,entrada,salida,decision

– ¿Como aseguraste que el flujo del diagrama fuera claro y fácil de seguir?

Por medio de las flechas.

– ¿Identificaste algún paso redundante o innecesario en tu diagrama? ¿Como lo optimizarías?

Si,simplificarlo seria con la verificación de los números primos

• Sobre la toma de decisiones y condiciones:

– ¿Como manejaste las condiciones en los ejercicios?

En un principio me parecía complicado comprender como desarrollarlas pero con el ejemplo de la profesora, me facilito la comprensión.

– ¿Que pasaría si el usuario ingresa un valor no valido (por ejemplo, un numero negativo)? ¿Como podrías mejorar tu diagrama para manejar estos casos?

Se podría agregar una condición adicional para manejar estos casos, devolviendo un mensaje de error o indicando que los números negativos no son primos.

• Descripción Sobre los bucles y repeticiones:

– ¿Que tipo de bucle utilizaste en los ejercicios (por ejemplo, while o for)? ¿Por que elegiste ese tipo?

El tipo while, se me facilito comprensión.

– ¿Como garantizaste que el bucle terminara correctamente sin caer en un ciclo infinito?

Con una condición para detenerse clara,como lo fue que el divisor fuera menor que n.

– ¿Podrías resolver el mismo problema sin usar un bucle? ¿Como lo harías?

No, creo que ya esta de la manera mas eficiente posible.

• Sobre la eficiencia y optimización:

– ¿Consideras que tu diagrama de flujo es eficiente en términos de pasos y recursos?

Si, trate de optimizarlos lo mejor posible en el menor numero de pasos necesarios.

– ¿Hay alguna forma de simplificar o reducir el numero de pasos en tu diagrama?

Si la hay la desconozco por completo.

– ¿Como podría escalar tu diagrama para resolver problemas mas complejos?

Se podría integrar más decisiones o procesos, dividiendo el diagrama en subprocesos.

• Sobre la aplicación practica:

– ¿En que situaciones de la vida real podría aplicar los conceptos que practicaste con estos ejercicios?

Tal vez en futuras programaciones en el transcurso del semestre.

– ¿Como podría adaptar tu diagrama de flujo para resolver un problema similar pero con diferentes requisitos?

Lo adaptaría según los requisitos requeridos.

– ¿Que ventajas tiene utilizar un diagrama de flujo frente a otras herramientas de planificación o programación?

Creo que permiten visualizar claramente la estructura de un algoritmo y tomar decisiones lógicas paso a paso.

• Sobre el aprendizaje y mejora continua:

– ¿Que fue lo mas desafiante de crear estos diagramas de flujo? ¿Como lo superaste?

Las asignaciones en lo personal,una vez que las logre especificar las condiciones eran mas sencillas implementar.

– ¿Que conceptos nuevos aprendiste o reforzaste con estos ejercicios?

El uso de los bucles,el como utilizarlos correctamente

– Si tuvieras que volver a hacer los ejercicios, ¿que cambiarias o mejorarlas?

Tal vez buscaría otras soluciones a los problemas para buscar optimizarlos.

• Sobre la comunicación y claridad:

– ¿Crees que otra persona podría entender fácilmente tu diagrama de flujo? ¿Por que?

Si,por que al final el diagrama de flujo es un medio el cual con los conocimientos mas básicos puedes comprender el algoritmo que se requiere lograr.

– ¿Como podría mejorar la claridad y legibilidad de tu diagrama?

Con mas experiencia y nuevos aprendizajes para simplificar procesos.

– ¿Que elementos adicionales (comentarios, anotaciones, etc.) podría agregar para hacerlo mas comprensible?

Agregar anotaciones o comentarios explicativos dentro del diagrama ayudaría a mejorar la comprensión.

Conclusión

Con la toma de decisiones, el uso de bucles y la estructura lógica de los algoritmos,pude identificar cómo optimizar y hacer más eficiente un proceso así como asegurarme de que los pasos fueran claros y fáciles de seguir en la mayoría de los ejercicios.

Al trabajar en un diagrama fue importante considerar no solo la exactitud de la solución sino también la claridad del algoritmo, para que cualquier persona pueda entenderlo independientemente de su experiencia en programacion. Las decisiones tomadas sobre cómo manejar condiciones y cómo estructurar los bucles ayudan para crear un proceso más eficiente y claro.

El proceso me permitió reflexionar sobre cómo los diagramas de flujo son herramientas visuales muy fáciles de seguir,no solo para programadores sino también para cualquier persona que quiera planificar un proceso de manera estructurada.