

Trabajo Práctico Nº0	Unidad 0
Modalidad: Semi -Presencial	Estratégica Didáctica: Trabajo individual.
Metodología de Desarrollo: Det. docente	Metodología de Corrección: Vía Classroom.
Carácter de Trabajo: Obligatorio – Con Nota	Fecha Entrega: A confirmar por el Docente.

ALUMNA:

DOMINGUEZ, SANDRA

Marco Teórico:

Responder el siguiente cuestionario en función de la bibliografía Obligatoria.

1. Explicar la cita docente

Parte I – Introducción a la Materia.

“Estudiaremos las estructuras de datos para aprender a escribir programas más eficientes”

Se hace hincapié en la estructura de datos principalmente porque se busca un priorizar la calidad del manejo de la información del sistema, dado que es lo más importante de resguardar o recuperar. Para ello se deberá comprender en profundidad el problema a resolver, ya que de esta manera es más factible dar con las estructuras correctas que permitirán una buena organización. Una vez lograda esta etapa satisfactoriamente, los algoritmos adecuados serán más sencillos de obtener.

2. Explicar los 2 enfoques presentados que deben cumplir el diseño de algoritmos.

Al crear un programa se busca contemplar estos principios:

- Diseñar un algoritmo que sea fácil de entender, codificar y depurar.
- Diseñar un algoritmo que haga un uso eficiente de los recursos de la computadora.

Enfoque de 3 pasos:

Este enfoque es adecuado para seleccionar una estructura de datos, consta de tres pasos y se centra en el proceso de diseño. Su preocupación se centra en los datos y lo que se hará con ellos, su representación y la implementación de esa representación. Consta de 3 preguntas que apuntan a:

- entender el problema.
- determinar las limitaciones de recursos.
- seleccionar la estructura de datos más adecuada.

Enfoque sistemático de cinco pasos:

- Restricciones:** se trate de entender y definir todas las limitaciones del problema. la entrevista se debe utilizar para comprender profundamente el problema y sus detalles, de lo contrario no sería adecuado intentar resolver el problema.

- b. **Generación de ideas:** a medida que se trabaje en un problema surgirán más preguntas y cuestiones, con la práctica se logrará fluidez para lidiar con estas cosas. Para resolver un problema desconocido deberíamos intentar:
 - a. simplificar el problema.
 - b. Probar la solución con algunos ejemplos.
 - c. Buscar la estructura de datos más adecuada al problema que intentamos resolver.
 - d. Buscar que soluciones se han dado a problemas parecidos al nuestro.
- c. **Complejidades:** es preciso poder calcular la complejidad del problema para poder elegir la solución más adecuada. se tendría que calcular la complejidad espacial y del tiempo, para poder conocer las compensaciones entre éstos a la hora de buscar la solución al problema a solucionar.
- d. **Codificación:** Si ya se han resuelto los 3 ítems anteriores (definición de las restricciones, generación de ideas y cálculos de la complejidad espacial y temporal) y se ha encontrado una solución adecuada, se puede pasar a la codificación. Es aconsejable escribir el código en pequeñas funciones en pos de que sea limpio y ordenado.
- e. **Prueba:** es una buena práctica revisar el código línea por línea con un caso de prueba. Se debería probar algunos casos de prueba.

Casos de prueba normales: son los casos de prueba positivos, contienen el escenario más común, y el foco está en el funcionamiento de la lógica básica del código.

Casos de borde: estos casos prueban los límites del código con casos extremos, sirven para lograr un código más robusto.

Prueba de carga: se prueba con una gran cantidad de datos, sirve para ver si el código es lento o necesita mucha memoria.

Antes de dar por terminado un código, deberían ejecutarse mentalmente todos estos casos de prueba.

3. ¿Qué se entiende por estructuras de datos?

“...una estructura de datos es cualquier representación de datos y sus operaciones asociadas.”

Son las distintas estructuras que se utilizan para representar la información en una computadora. En este sentido, una estructura de datos es una forma de organizar un conjunto de datos elementales, de manera eficiente, con el objetivo de facilitar su uso y manipulación.

4. ¿Por qué debemos medir la eficiencia de los Algoritmos?

La solución de un problema va a pasar por la elección y desarrollo correcto de los algoritmos, porque para lograr un sistema que funcione necesito primeramente del algoritmo que luego será codificado en un lenguaje de programación y ejecutado en una computadora.

Dado que un algoritmo es una secuencia de pasos para resolver un problema, pueden existir muchas soluciones para un mismo problema. En pos de la eficiencia del código debemos elegir el algoritmo más eficiente, comparando las diferentes soluciones que proponen y tomando en cuenta la cantidad de líneas de código, el tiempo de ejecución, la cantidad de pasos., la cantidad de

recursos que utiliza. La complejidad algorítmica estudia la eficiencia teórica del programa mediante operaciones matemáticas.

5. ¿Por qué hablamos de una “Filosofía” de las Estructuras de Datos?

Generalmente una solución se considera eficiente si consume menos recursos y es más veloz que otras. se debe analizar el problema en función del rendimiento para lograr la elección de la estructura de datos adecuada al problema, y no elegir tal estructura en función de lo familiarizado o cómodo que se sienta el desarrollador con ésta, de lo contrario se obtendrá un programa lento, donde se complejizará otras áreas para lograr los objetivos propuestos.

6. ¿Por qué relacionamos las Estructuras de Datos con los TDA?

Una estructura de datos es una forma de organizar un conjunto de datos elementales, de manera eficiente, y un tipo de dato abstracto es un conjunto de datos u objetos creado de manera personalizada por un programador para un fin específico, es una abstracción que permite modelar las características de un elemento en particular que provee una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de como estén implementadas dichas operaciones, esto permite que un mismo TDA puede ser implementado utilizando distintas estructuras de datos y proveer la misma funcionalidad.

7. Profundizar la Frase: “El costo de la Solución es la Cantidad de Recursos que la Solución Consume”

Si bien el costo se mide habitualmente en función del tiempo que invierte una solución para cumplir los objetivos fijados. La frase se relaciona con la eficiencia de un programa, donde no sólo se tiene en cuenta el cumplimiento de los objetivos, sino también la cantidad de recursos que utiliza para cumplirlos.

8. ¿Cuáles “Deberían” ser los Pasos para seleccionar una Estructura de Datos?

Este enfoque es adecuado para seleccionar una estructura de datos, consta de tres pasos y se centra en el proceso de diseño. Su preocupación se centra en los datos y lo que se hará con ellos, su representación y la implementación de esa representación. Consta de 3 preguntas que apuntan a identificar;

- a. entender el problema.
- b. determinar las limitaciones de recursos.
- c. seleccionar la estructura de datos más adecuada.

9. Relacionar estructuras de Datos Vs. Costos y Beneficios?

La elección de la estructura de datos debe ser elegida en función de un cuidadoso análisis del problema a resolver, esto dependerá de los requerimientos de espacio y velocidad de acceso por ejemplo, dado que no todas las estructuras responderán de forma igualmente eficiente a cada requerimiento y/o limitación del diseño.

10. Describir por que las Estructuras de Datos tienen una estructura “Lógica” y una “Física”, argumentar.

Los tipos de dato tienen forma física y lógica. La forma física es cuando se implementa el tipo de dato, y la definición su forma lógica. Si tengo una clase determinada, la parte lógica serán las funciones públicas, sus entradas y salidas.

11. Relacionar las Estructuras de Datos con Patrones.

Los patrones de diseño están pensados para simplificar y ahorrar trabajo. Mediante los patrones se busca generalizar conceptos de diseño para dar solución a problemas que se repiten, es una forma de transferir conocimientos desde los programadores más experimentados a otros que no lo son tanto, ofreciendo una solución ya probada, de todas formas no se puede perder de vista el hecho de que no todos los problemas son iguales, y por lo tanto los patrones de diseño deben ser elegidos muy a consciencia, habiendo comprendido en profundidad el problema a solucionar. En este sentido se parecen a las estructuras de datos, es decir, cada uno tiene sus pros y sus contras, lo cual exige análisis y conocimiento.

12. Armar una Relación Narrativa que incluya: Problema, Algoritmo y Programa.

Un problema es algo que requiere una solución, en el caso de la programación se trata de algo que se espera que se ejecute en una computadora y dé respuesta a determinada necesidad. Primeramente se debe analizar y comprender cabalmente el problema, y tomar en cuenta las restricciones que puedan existir. Su solución puede pensarse mediante un algoritmo, donde se tomara en cuenta todos los pasos necesarios para resolver algoritmos, pueden existir más de un algoritmo que resuelvan el mismo problema, en ese caso, se compararán y se elegirá el que sea más eficiente. Al implementar el algoritmo en algún lenguaje de programación se obtendrá un programa.