

Informática: Ciencia que estudia la resolución de problemas, del mundo real a través de la computadora.

Etapas en la resolución de problemas:

1. Análisis: se analiza el problema en su contexto del mundo real, intentando simplificarlo al máximo posible, tanto como los datos a utilizar.
2. Diseño: buscar una solución modularizada, para reducir trabajo, distribuirlo, y poder re-utilizar los módulos.
3. Implementación: escribir los algoritmos, representando correctamente los datos.
4. Verificación: verificar que la solución sea la deseada.

Algoritmo: conjunto de instrucciones y operaciones, interpretadas por un autómatas buscando una solución deseada en un tiempo finito.

Programa: conjunto de instrucciones ejecutables sobre una computadora, en lenguaje de programación, que permite cumplir una función concreta. Se ejecutan en un tiempo finito. Programa = instrucciones + datos.

Instrucción: representa las operaciones que ejecutara la computadora al interpretar el programa.

Dato: representación del mundo real que se utilizan como valores que interpreta el programa.

Tipo de dato: es una clase de objetos ligados a ciertas operaciones para crearlo y modificarlo. Tienen un rango de valores posibles, un conjunto de operaciones posibles realizables.

Estructura de control: conjunto de instrucciones que puede modificar el flujo de ejecución del programa.

1. Secuencial: flujo en el cual una instrucción, sigue a otra.
2. Iterativa: se ejecutara sin saber el número de veces que lo hará, puede ser post-condicional, pre-condicional, o repetitiva.
 - Pre-condicional: primero evalúa la condición, y si es verdadera se ejecuta el bloque de instrucciones.
 - Post-condicional: se ejecutara al menos una vez, evaluando al final del bloque si volverá a ejecutarse.
 - Repetitiva: se ejecuta sabiendo el número de veces que realizara el bloque de instrucciones.
3. Decisión: toma de decisiones, en función de una condición.

Modularización: dividir el problema en subproblemas, buscando legibilidad, eficiencia y un algoritmo más limpio. Cada módulo realiza una tarea específica y simple, facilitando el flujo de nuestro algoritmo, y deben ser independientes entre sí, dichas tareas deben

combinarse para resolver el problema original. Esto se llama diseño top Down. Los módulos pueden ser procedimientos, o funciones. Y se comunican a través de parámetros, o variables globales, aunque esto último no es recomendable.

- Procedimiento: conjunto de instrucciones, que se encarga de resolver una tarea específica, realizando las operaciones correspondientes. Devuelve 0, 1 o más valores de cualquier tipo.

Función: conjunto de instrucciones que realiza una tarea específica, y como resultado retorna un único valor del tipo simple.

Variables locales y parámetros.

- Ocultamiento de datos: Los datos exclusivos de un módulo, no son visibles y accesibles por otros módulos. Éstos se declaran locales al mismo.
- Parámetros: Son datos compartidos que se transmiten entre módulos, y son declarados como parámetros.

Comunicación entre módulo y programa.

- Variables globales: no se recomienda por distintos motivos.
 1. Demasiadas variables declaradas.
 2. No especifica que variable corresponde a cada módulo.
 3. Se pueden superponer los nombres usados.
 4. Al no estar ocultas, se pueden modificar involuntariamente.
- Uso de parámetros: los datos compartidos se deben especificar como parámetros que se transmiten entre módulos. Se puede realizar por valor o referencia:
 1. Un parámetro por valor es un dato de entrada el cual el modulo recibe una copia de un valor proveniente de otro modulo o el principal.
 2. Un parámetro por referencia es un dato que consiente una dirección de memoria donde se encuentra la información compartida con otro modulo o programa.

Estructura de datos: es un conjunto de variables relacionadas entre sí, que se pueden operar como un todo, bajo un único nombre. Representan datos del mundo real. Dichas estructuras se clasifican teniendo en cuenta cómo son sus elementos, cómo se accede a ellos, cómo va a ser su ocupación en memoria y cómo se relacionan sus elementos.

- Registro: estructura que permite agrupar diferentes tipos de datos.
 1. Ocupa posiciones contiguas en la memoria (lineal).
 2. Es de acceso directo a sus campos desde una variable referenciada
 3. Es de almacenamiento estático.

Dentro de la estructura puede almacenar varios tipos de datos, especificándolos.

- Arreglo: colección de elementos que se guardan consecutivamente en la memoria, las cuales se referencian por índices. Cuenta con una dimensión física, la cual determina la ocupación en memoria. Y la dimensión lógica indica la cantidad de posiciones de memoria ocupadas, cargadas desde la primer posición, en forma consecutiva.
 1. Todos los elementos, deben ser del mismo tipo.
 2. Se puede acceder a cualquier parte del arreglo mediante acceso directo referenciando su índice.
 3. La memoria durante la ejecución es fija.
 4. Siempre tiene un elemento predecesor, y un sucesor.
- Lista: colección de elementos homogéneos, con una relación lineal que los vincula. Cada elemento tiene un único predecesor, excepto el primero, y un único sucesor, excepto el último. Los elementos que la componen no tienen ocupación lineal en memoria, pueden estar dispersos por la memoria, pero mantienen un orden interno.

Calidad de un programa

- Corrección: el grado en que un programa resuelve el problema, y consigue los objetivos encomendados del cliente.
- Fiabilidad: el grado en el cual se espera que el programa realice las operaciones especificadas, y con precisión.
- Eficiencia: la cantidad de recursos hardware y software que necesita el programa para realizar las operaciones en tiempos adecuados.
- Integridad: el grado con el cual puede controlarse el acceso a software, o a datos no autorizados.
- Facilidad de uso: el esfuerzo que conlleva a un buen uso del programa.
- Facilidad de mantenimiento: el esfuerzo requerido encontrar y reparar errores.
- Flexibilidad: el esfuerzo requerido para modificar el programa en funcionamiento.
- Facilidad de prueba: el esfuerzo requerido para probar el programa para que cumpla con lo especificado.
- Portabilidad: el esfuerzo requerido para transferir el programa a otro hardware o sistema operativo.
- Reusabilidad: el grado en que las partes de un programa, pueden usarse en otro.
- Interoperabilidad: el esfuerzo necesario para que nuestro programa, se comunique con otros.
- Legibilidad: el código debe ser fácil de leer y entender.
- Documentación: todo nuestro código, debe estar documentado mediante texto, o gráficos para favorecer la comprensión.

Técnicas para la corrección del programador:

- Test: proceso mediante el cual se proveen evidencias respecto al trabajo que realiza el programa.

- Debugging: proceso para localizar el error.
- Walkthroughs: consiste en recorrer el programa mediante una audiencia.
- Verificación: proceso de analizar las postcondiciones en función de las precondiciones.

Eficiencia: calidad de los algoritmos, basada en la utilización óptima de los recursos, donde se ejecutara el programa. Se intenta usar la menor memoria posible, en el menor tiempo.

Métodos para medir tiempo de ejecución:

- Análisis empírico: ejecutar el programa y cronometrar el tiempo transcurrido, midiendo el tiempo de ejecución. La ventaja de este método, es que no hay que tener conocimiento para poder utilizarlo. Y la desventaja, es que no es el tiempo real de ejecución por que depende de los recursos de la maquina a analizar. Se tiene que ejecutar varias veces.
- Análisis teórico: el otro método consiste en abrir el código, e ir sumando línea a línea los tiempos de ejecución de las instrucciones. Este método es más complicado, pero es el tiempo real de ejecución de nuestro algoritmo. Se toman en cuenta las asignaciones, operaciones fundamentales, y comparaciones.